Greenlee 560 Summary 2

Elliot Greenlee

February 2, 2017

1 Introduction

This paper explains an implementation of remote procedure calls for network communication. The core concept of remote procedure calls is using the commonly used and well known paradigm of procedure calls for communication. A call is made, execution occurs on the other computer, and data is sent back in much the same way as the local example.

2 Problems

What problems does this paper try to solve? Why are such problems important?

Building distributed and networked systems is a difficult task, even for researchers and developers with significant experience. The largest constraint on this task is the lack of existing communication tools. This problem is important to solve as the trend towards distributed systems continues. Distributed systems are necessary, and cannot be developed and optimized without proper communication tools.

3 Assumptions

What are assumptions made by this paper? Are they verifiable? Are there logical holes in such assumptions?

This paper assumes that distributed system communication is an unnecessary, rather than fundamental, development difficulty. It would seem that in environments where separate hardware is connected by communication, such communication is a fundamental difficulty.

The writers assumed that emulating system procedure calls is best development core concept. However, this caused problems with missing common features like time out, and with not alerting users to deadlocks and loops on the callee machine.

4 Solutions

What are the major solutions of this paper? Do you think the solutions in this paper will work for the problem? Do you think the paper evaluates the solutions in a convincing manner? List two limitations of the solutions proposed in this paper, and outline your method to fix them.

The designers created a high level remote procedure call communication protocol to assist in distributed system development. Additionally, they created their own transport protocol which minimized the real-time costs of messaging. It appears that this solution has been very successful; it has inspired networked file access. However it also seems this solution has many limitations, so it is surprising to me that it was successful.

4.1 Limitations

The developers chose to create a high level, non extensible system, but this keeps skilled users from creating the things they would like to have on top of the framework, such as a time out mechanism or an explicit termination protocol. As such, neither broadcasting nor multi-casting can be built on top of the increased efficiency they developed. Making extensibility on top of the efficiency one of the major design tenets could allow skilled users to improve the system. However, they mostly designed the system for use in-house, so perhaps their decision makes sense.

The main protocol for communicating is one call, and one return of results. This is a direct result of attempting to maximize efficiency. If the callee code has a looping or deadlocked execution, there is no information sent to the caller, and no upper bounded wait. The paper only hints at a solution where the callee monitors its own progress and sends back a message, which would seem to be an excellent solution to this problem.