Forty Years of Pretending

@reiddraper

About me

Eng. @ Basho

@reiddraper

Nobody ever got **fired** for making an **RPC** call

RPC claims **transparency** between local and remote procedure calls

This abstraction fails

Even a simple web app is a distributed system

A resurgence with Javascript

RPC



Pretending



Tomorrow?

RPC

Remote Procedure Call

[...] RPC mechanism is one in which local procedures and remote procedures are (effectively) indistinguishable to the programmer. - Nelson [...] the calling environment is suspended, the parameters are passed across the network to the environment where the procedure is to execute [...], and the desired procedure is executed there. [...] the results are passed backed to the calling environment, where execution resumes as if returning from a simple single-machine call

- Nelson

A synchronous network call does **not** make RPC

send(2) is not an RPC call

transparency between local and remote procedure calls

Motivation for RPC

Distributed systems are hard

The procedure call is familiar

Let's lift distributed programming to the procedure call **abstraction** Let's lift distributed programming to the **method** call abstraction Let's lift distributed programming to the **function** call abstraction

History

Telnet

RFC 674 **1974** Postel and White

Procedure Call Protocol

RFC 707 **1976** *White*

Remote Procedure Call

The principal goal of all resourcesharing computer networks, including the now international ARPA Network (the ARPANET), is to usefully interconnect geographically distributed hardware, software, and human resources

- White

It forces upon the user all of the trappings of the resource's own system. - White

It provides no basis for bootstrapping new composite resources from existing ones. - White


An alternative history

The Actor Model 1973 Hewitt

RFC 684 1975 Schantz

[...] objection to the"PCP philosophy"

- Schantz

RPC blurs the line between **local** (cheap) and **remote** (expensive) RPC blurs the line between local (cheap) and remote (expensive) [...] a model which relies on procedure calling for its basis does not take into account the special nature of the network environment

- Schantz

[...] and that such an environment can be more suitably handled in a message passing model.

Schantz

Goes on...

A Critique of the Remote Procedure Call Paradigm 1988 Tanenbaum van Renesse Implementing Distributed Systems Using Linear Naming 1993 Bawden

RFC 2616 (HTTP 1.1) **1999** Fielding et al.

REST 2000 Fielding

Problems with RPC

Consistent code

A new version of the application is required. Please refresh the page. Reload

How do you atomically update the code?

Handling failure

What if the remote node never responds?

Or the underlying TCP connection is closed?

Did it fail halfway through?

Did it fail right before it returned to us?

Why don't we have a stacktrace?

What if our procedure made another RPC call?

Serializing values

How do you serialize a filedescriptor?

How do you serialize a database connection?

The streaming problem

Can we start processing partial results?

The continuation problem

The procedure call only models A->B->A communication

(8) Fallacies of Distributed Computing

Fallacies of Distributed Computing

1. The network is reliable

2. Latency is zero

3. Bandwidth is infinite
4. The network is secure

5. Topology doesn't change

6. There is one administrator

7. Transport cost is zero

8. The network is homogeneous

Today

Javascript and compile-to-Javascript

Code sharing!



A new version of the application is required. Please refresh the page. Reload

Even a simple web app is a distributed system

meteor.js derby yesod-fay shoreleave hapstack dnode now.js nodejs-light_rpc

Tomorrow

The principal goal of all resourcesharing computer networks, including the now international ARPA Network (the ARPANET), is to usefully interconnect geographically distributed hardware, software, and human resources

- White

Distributed systems are still hard

Implementing Distributed Systems Using Linear Naming 1993 Bawden

Bloom (UC Berkeley)

Questions

@reiddraper