

Remote Procedure Call

Semplifica comunicazione client server tra due terminali diversi.

A tempo di scrittura - il programmatore deve includere le interfacce che son necessarie per usare le RPC

A tempo di esecuzione - è necessario un supporto che permetta il funzionamento delle RPC

A tempo di compilazione - Aggancia al mio codice linee di codice che permette di trasformare i dati di modo che il server possa interpretare i miei dati. Rende più semplice lo scambio di parametri .

Stack Protocollare divisa in tre livelli: BLAST , CHAN , SELECT

Select = in una singola macchina ci possono essere più chiamate a RPC , il livello select selezione qual'è la giusta RPC a cui inviare i parametri per ogni richiesta. =MULTIPLEXING

Chan = sincronizza le richieste e le risposte

Blast = le richieste con i parametri possono essere pesanti e quindi in questo livello vengono trasformati in frame da 32 pacchetti ripetute un numero di volte necessario per trasferire il messaggio. A ogni frame si riceve un ACK.

Se un pacchetto non è ricevuto ricevuto correttamente allora lo reinvia con timeout aumentato esponenzialmente ogni volta. Con meccanismo slow start..

DISPATCHER(Select) - seleziona la procedura appropriata per la richiesta, può essere di due tipi:

FLAT - assegna id per ogni procedura

GERARCHICA : program + procedura ?

Presentation Formatting: INTEROPERABILITA'

Chiamante e Chiamato son su macchine distinte e quindi con architetture distinte.

Il passaggio dei parametri deve essere regolato ..

I numeri: 2 problematiche -

1) Gli interi rappresentati in little o big endian??

2) Esiste una versione standard Floating Point ma non tutti la seguono

Strategie di conversione

1) Si crea un modello di informazione intermedio e più astratto al quale tutti i calcolatori si riferiscono. Vengono chiamate forme canoniche e sono una sorta di lingua intermedia per far dialogare i diversi calcolatori

2) Receiver makes right = Il chiamante manda info , il receiver è in grado di stabilire secondo quale forma canonica sono quelle info. Ogni calcolatore ha quindi diverse forme canoniche a cui fare riferimento e attraverso alcuni bit individua la forma canonica utilizzata dal sender. Svantaggio = se io receiver non conosco la forma che il sender utilizza= non possiamo dialogare

Una forma canonica può trasferire i dati in modalità TAGGED e UNTAGGED

TAGGED = metto nell'header del pacchetto informazioni riguardo a ciò che ti sto inviando = tipo di dato + lunghezza

UNTAGGED = dati per cui si presuppone ci sia un accordo tra sender e receiver riguardo a tipologia e dimensione

Una forma canonica intermedia popolare= eXternal DATA Representation
= utilizza la forma TAGGED