

Re: Remobjects, good in multi-threading environment?

Re: Remobjects, good in multi-threading environment?

Source:

<http://coding.derkeiler.com/Archive/Delphi/borland.public.delphi.thirdpartytools.general/2007-12/msg00317.html>

- *From:* "Kim Madsen" <kbm@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Fri, 7 Dec 2007 16:05:18 +0100
-

Hi Arthur,

There are some pretty extreme requirements that have to be met in our application. We're talking about the transmission of "realtime" numerical data here (with a 1-s or 2-s sample interval). The worst-case scenario that must still work is a connection with a latency of 2 s and a bandwidth of only 240 bytes/sec (more specifically, this is a Thrane & Thrane Inmarsat mini-M satellite terminal). Add to this the complication that the connection is interrupted regularly. In case of an interruption, it is mandatory that as soon as the connection is re-established, the most recent data is transferred first. Simultaneously, the "gaps" in the client's database must be filled in in reversed sequence, but "current" data must be transferred at all time because we may lag behind no more than 5 seconds.

This can certainly be handled by the WIB. It contains automatic reconnection facilities and even failover and loadbalancing (in those cases where that is needed, which may not be your specific requirement).

The messages can be compressed to quite small amounts of data.

This document shows how our traditional transport stream (request/response) is formatted.

http://www.components4programmers.com/downloads/kbmmw/documentation/The_transport_stream_format.pdf

From it you can see that an empty, non compressed request frame takes up

roughly 53 bytes.

That request however also expects a response back from the server, which again is roughly 53 bytes minimum.

Our WIB transport is using the same format, but includes a WIB header that contains some QoS etc. parameters required by the messaging features.

Hence that would be approx 91 bytes per uncompressed message, but would not require a response back.

Re: Remobjects, good in multi-threading environment?

Re: Remobjects, good in multi-threading environment?

Hence your worst case scenario would still work and we would be able to deliver one to five slim messages per sec (depending on compression levels etc).

The order of how the messages go thru can be handled by prioritizing messages that you want to go thru first.

best regards

Kim Madsen

kbm@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

www.components4developers.com

.