Network Working Group Michael Beeler (BBN-TEN
EX)
Request for Comments: 685 April 16, 1
975
NIC #32298

Response Time in Cross-network Debugging

Cross-network debugging is a means whereby a programmer at one computer on a network can debug a program which executes on another computer. One form of cross-net debugging has been in use for some years by programmers who maintain IMPs on the ARPA network. Another form has been used by ARPA network users who employ TELNET or RSEXEC to log into a distant computer and remotely run a debugger on that machine. In both of these cases, the debugger is almost entirely resident in the same machine as the subject program, and only a remote means of access to that computer distinguishes the activity from single-computer debugging.

In our case, we use a PDP-10 to perform complex debugging manipulations. Simple manipulations, and complex actions which the PDP-10 has partially digested into simple actions, are sent over the ARPA network to a PDP-11. The portion of the debugger resident in the PDP-11, where the subject program executes, can perform only simple actions (examine, deposit, start, stop, set breakpoint, etc.). This division of debugging computation between the two machines is implemented and in use at BBN. A user s manual is available (as (BBN]<DOCUMENTATION>XNET.DOC) describing the debugger s features and discussing some of the issues involved.

The purpose of this RFC is to describe our experience with response times the debugger exhibits. Response time is a serious problem in any elaboration to a debugger. Here we wish to discuss the contribution of communicating over the ARPA network to response time. The debugger (X-NET) keeps statistics during each debugging session, and a debugger command prints them out. We used a "standard scenario" to measure response times on two occasions. The first was debugging a PDP-11 at BBN on the same IMP as the PDP-10. The second was with a PDP-11 at SRI-ARC in California, with at least nine IMPs intervening.

BBN (local) SRI (distant)
TENEX LOAD AVG
START 6.0 4.65
FINISH 3.9 6.6
TIME OF DAY 15:30 EST 11:00 EST
DAY 4/10/75
4/11/75

Each session lasted about 10 minutes. The terms used below are:

message -- a single message generated by the PDP-10 portion of X-NET

active command -- a command which involves, actually or virtually, an interaction with the subject program (e.g., examine, deposit, start, stop, set breakpoint, etc.)

bytes -- the total of all (8-bit) bytes, both sent and received, plus any bytes due to receipt of asynchronous replies (e.g., breakpoint hit), during processing of the associated message or active command.

wait -- total elapsed time from handing message to implementation language (BCPL) network routines, to receipt of the reply from these routines and through an inferior process in X-NET

The 35 active commands in the scenario are:

1 load program
8 start or proceed program
3 halt program
16 examine contents of memory word
1 deposit new contents in memory word
1 set breakpoint
1 remove breakpoint
1 word search
1 copy program onto disk file
2 network/process management (see user’s manual)

The summary statistics are:.

BBN (local) SRI (distant)
AVG STD DEV AVG STD DEV
PER MESSAGE:
BYTES 154 295 164 295
WAIT 1.75 2.04 1.89 0.78 SEC
PER ACTIVE COMMAND:
MSGS 0.91 0.69 0.91 0.69
BYTES 150 331 150 331
WAIT 1.60 2.36 1.73 1.35 SEC

The standard deviation is relatively large partly because of a small number of samples, but even more because the message size and the command complexity are bimodal, as shown by the histograms below. The load and word search commands transferred many bytes, as did an examine (the first one while the program was halted; subsequent examines were answerable from the cache; see user s manual). Other commands needed little or no network transaction. Those which needed none at all produced a no-delay mode in the distribution of waiting time per active command.

We conclude that the delay due to network communication is tolerable. It is of the same order of magnitude as that often experienced on moderately loaded time sharing systems. More explicit measurements of delays seen by user programs in general are in progress at BBN and elsewhere; it is beyond the scope of this RFC to discuss these delays in detail, or to break down their causes into process activation, queueing, IMP performance, etc. Instead, we merely note that cross-network debugging is possible in a practical sense.

	PER MESSAGE	PER ACTIVE COMMAND
0	.	XXXXXXXXX
16	.	.
32	XXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXX
64	.	XXX
128	.	.
256	XX	.
512	XXXX	XX
1024	.	XX

SIZE (BYTES), BBN (local data) = SRI (distant data) Left column gives lower bound (inclusive) on logarithmic scale. Thus, two messages had at least 256 bytes but less than 512 bytes. An examination of network traffic per active command shows that it is actually trimodal: some commands are answered from the cache, incurring no network traffic; some, such as start or stop, require only a few tens of bytes; and some commands, such as word search and load, cause transfer of thousands of bytes.

	PER MESSAGE	PER ACTIVE COMMAND
0	.	XXXXXXXXX
1/16	X	.
1/8	.	.
1/4	X	.
1/2	XXXXXXXXXX	XXXXXXXX
1	XXXXXXXXXXXXXX	XXXXXXXXXXX
2	XXXX	XXXX
4	X	X
8	X	XX

WAITING TIME (SEC), BBN (local data)

	PER MESSAGE	PER ACTIVE COMMAND
0	.	XXXXXXXXX
1/16	.	.
1/8	.	.
1/4	X	.
1/2	XXXXXX	XXX
1	XXXXXXXXXXXX	XXXXXXXXX
2	XXXXXXXXXXXXX	XXXXXXXXXXXX
4	.	XX
8	.	.

WAITING TIME (SEC), SRI (distant data)