One-shot Acnet Requests

Timing analysis for RETDAT

Wed, Dec 26, 2001

To support Acnet data acquisition, the RETDAT protocol is used. During Big Save operations, in which all devices registered in the Acnet database are acquired for archiving purposes, an effort is made to accomplish this in the most efficient manner. This note documents examples of measured timing performance for such data using the newly-installed PowerPC-based Linac front-ends.

Example measurements

Test node0509 was used as a client to send various size Acnet RETDAT requests via the Linac server node0600 targeting equal numbers of devices from node0611, node0613, and node0614. Requests sized for 273, 84, and 3 devices were measured. In each case, the device data sought consisted only of 2-byte analog reading values. The example of 273 devices required IP fragmentation into three ethernet fragments, as each Acnet device specified in the request uses 16 bytes. The example of 84 devices required no fragmentation, but it nearly filled an ethernet frame. The example of 3 devices is a very minimal request. In each case, the client sends a single RETDAT request to node0600, which immediately forwards the entire request to a multicast address that reaches all Linac front end nodes. Each node receiving the request that includes at least one of its own devices arranges to build a single reply message in response. (Node0600 receives this same forwarded request, too, but since it finds none of its own devices therein, it ignores it.) When node0600 receives the last of the ensuing three responses, it arranges the data into a single reply and sends it to the client.

Two diagnostics, implemented via the system data stream support, were used to analyze what happened during the testing. One is the network frame diagnostics, which is a log of datagrams for the PowerPC/vxWorks nodes. These diagnostics show the following results, with all times in ms:

Network activities	273	84	3
Request arrival at node0600	36	25	20
Reply sent from node0600	44	29	21
Turnaround at node0600	8	4	1
Request arrival at node0613	42	27	20
Reply sent from node0613	43	28	21
Turnaround at node0613	1	1	1

The network diagnostics do not indicate the timing of arrival or departure of the IP fragments necessary to contain the request datagram. At this level, they are significant, because a full ethernet frame requires 1.2 ms of bandwidth at 10 Mbs. Beginning from the arrival of the first fragment, the turnaround time for the 273 device case at node0600 would be about 11 ms, assuming that the three fragments are essentially contiguous, which is quite likely in the CAT-5 environment we use here.

The other diagnostic logs all task activities. These occur at a very high rate, causing the circular buffer to wrap in a fraction of a second, so one needs a fast finger to capture the relevant data. In order to make it easier to find the task activities of interest here, the client was synchronized to begin its 15 Hz activity at 17 ms rather than the usual 3 ms past the Booster reset event time. Here is a brief summary of the task activity found in this way:

Task activities	273	84	3
at node0600			
First fragment rec'd	34	25	20
Forward to multicast	37	26	20
First node reply	44	28	21
Reply sent	45	29	22
Turnaround	11	4	2
Number task executions	33	30	27
Total cpu time	2.24	1.73	1.43
Idle time	8.54	2.86	0.53
at node0613			
First fragment rec'd	40	27	20
Reply sent	43	28	20
Turnaround	3	1	<1
Number task executions	11	8	8
Total cpu time	0.87	0.56	0.43
Idle time	2.30	0	0

Compared with the network diagnostics, the turnaround time for the 273-device case is longer at `node0600`, because the timing of the fragment arrivals is observable. The idle time in `node0600` includes both the time available between fragment arrivals and the time awaiting the replies from the contributing nodes, some of which includes the time for the bandwidth of the forwarded request datagram. The idle time in `node0613` can only include the unused time between fragment arrivals.

The number of task executions is large for the server node, because it must provide both server and non-server support for the same request, even though it is not a contributing node in these examples. For such noncontributing nodes, the time needed to accept and process (and ignore) the request was 0.42, 0.30, and 0.27 ms, respectively.

All task timing is made by sampling the time, using the PowerPC Time Base Register, once every time vxWorks invokes the task switch hook routine. This means that the measured task execution times include all task-switching overhead, including the time used to record such diagnostics. The minimum task execution time observed seems to be about 0.009 ms.

These examples involved requests for analog readings, which requires only very simple support. Other examples can be made that may require more time to support, of course. But even these simple requests involve access to nonvolatile memory, because the analog readings are housed in fields in the `ADATA` table that resides in nonvolatile memory. Each such access costs about 1 microsecond; hence, the time to access nonvolatile memory for the case of 273 devices, in which 91 are to be found therein, is about 0.09 ms, or about 10% of the total cpu time to support this request at `node0613`.