How to Periodically Updating Labview chart when collecting multi channel data at a high rate

Asked 16 years ago Modified 16 years ago Viewed 12k times

1

Looking for some help with a Labview data collection program. If I could collect 2ms of data at 8kHz (gives 16 data points) per channel (I am collecting data on 4 analog channels with an National Instruments data acquisition board). The DAQ-MX collection task gives a 1D array of 4 waveforms.

If I don't display the data I can do all my computation time is about 2ms and it is OK if the processing loop lags a little behind the collection loop. Updating the chart in Labview's front panel introduces an unacceptable delay. We don't need to update the display very quickly probably at 5-10Hz would be sufficient. But I don't know how to set this up.

My current Labview VI has three parallel loops

1. A timed-loop for data collection
2. A loop for analysis and processing
3. A low priority loop for caching data to disk as a TDMS file

Data is passed from the collection loop to the other loops using a queue. Labview examples gave me some ideas but I am stuck.

Any suggestions, references, ideas would be appreciated.

Thanks

Azim

Follow Up Question

eaolson suggests that I re-sample the data for display purposes. The data coming from the DAQ-MX read is a one dimensional array of waveforms. So I would need to somehow build or concatenate the waveform data for each channel. And then re-sample the data before updating the front panel chart. I suppose the best approach would be to queue the data and in a display loop dequeue the stack build and re-sample the data based on screen resolution and then update the chart. Would there be any other approach. I will look on (NI Labview Forum)[http://forums.ni.com/ni/board?board.id=170] for more information as suggetsted by eaolson.

Updates

1. changed acceptable update rate for graphs to 5-10Hz (thanks Underflow and eaolson)
2. disk cache loop is a low priority one (thanks eaolson)

3. Thanks for all the responses.

labview daq-mx

Share

Improve this question

Follow

edited Dec 6, 2008 at 18:22

asked Dec 6, 2008 at 4:16

Azim J

8,290 ● 7 ● 41 ● 61

2 Answers

Sorted by: Highest score (default)

5

Your overall architecture description sounds solid, but... getting to 30Hz for any non-trivial graph is going to be challenging. Make sure you *really need* that rate before trying to make it happen. Optimizing to that level might take some time.

References that should be helpful:

You can defer panel updates. This keeps the front panel from refreshing until you're ready for it to do so, allowing you to buffer data in the background, and only draw it occasionally.

You should know about (a)synchronous display. This option allows some control over display rates.

There is some general advice available about speeding execution.

There is a (somewhat dated) report on execution speed on the LAVA forums. Googling around the LAVA forums is a great idea if you need to optimize your speed.

Share Improve this answer

Follow

answered Dec 6, 2008 at 5:25

Joe Z

752 ● 2 ● 10 ● 29

The article you've linked to under 'general advice' is worth studying carefully as it covers a lot of the issues and pitfalls around execution speed in LabVIEW. – nekomatic Dec 8, 2008 at 15:28

thanks for the links. of course I did originally over estimate the refresh rate. – Azim J Dec 13, 2008 at 20:52

2

Television updates at about 30 Hz. Any more than that is faster than the human eye can see. 30 Hz should be at the maximum update rate you should consider for a display, not the starting point. Consider an update rate of 5-10 Hz.

LabVIEW charts append the most recent data to the historical data they store and display all the data at once. At 8 kHz, you're acquiring at least 8000 data points per channel per second. That means the array backing that graph has to continuously be resized to hold the new data. Also, even if your graph is 1000 pixels across, that

means you're displaying 8 data points per screen pixel. There's not usually any reason to display any more than one data point per pixel. If you really need fast update rates, plot less data. Create an array to hold the historical data and plot only every Nth data point, where N is chosen so you're plotting, say, only a few hundred points.

Remember that your loops can run at different rates. It may be satisfactory to run the write-to-disk loop at a much lower frequency than the data collection rate, maybe every couple of seconds.

Avoid property nodes if you can. They run in the UI thread, which is slower than most other execution.

Other than that, it's really hard to offer a lot of substantial advice without seeing code or more specifics. Consider also asking your question at the NI LabVIEW forums. There are a lot of helpful people there.

Share Improve this answer

Follow

answered Dec 6, 2008 at 16:17

eaolson

15.1k ● 7 ● 45 ● 60

Of course if you're doing things to the UI, you may need to use property nodes - your concern then should be to make sure that they are not blocking your performance-critical code. – nekomatic Dec 8, 2008 at 15:30