# artdaq Utilities - Idea #23029

## DAQInterface could launch processes in parallel across nodes

07/31/2019 02:18 PM - John Freeman

| | | | |
|---|---|---|---|
| **Status:** | New | **Start date:** | 07/31/2019 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | John Freeman | **% Done:** | 0% |
| **Category:** | | **Estimated time:** | 0.00 hour |
| **Target version:** | | **Spent time:** | 0.00 hour |
| **Experiment:** | - | | |

**Description**

Something that came up during Monday's artdaq meeting: when we're using direct process management, during the boot transition DAQInterface sequentially launches processes node-by-node. E.g., if it's booting artdaq processes on both mu2edaq01 and mu2edaq11, it'll first launch the mu2edaq01 processes, and only then launch the mu2edaq11 processes (or vice versa). It's probably worth looking into the possibility of having DAQInterface launch processes across nodes at the same time, as this may speed things up. Of course, additional thought will need to be given to things like error messages (e.g., if DAQInterface can't launch the processes on either node, you don't want colliding error messages concerning each failure cluttering up the screen).

**Related issues:**

| | | |
|---|---|---|
| Related to artdaq - Support #23764: artdaq boot sequence takes significant ti... | **New** | **12/13/2019** |

### History

**#1 - 08/03/2019 01:07 PM - John Freeman**

tl;dr : going parallel is not working out as hoped.

I've been running tests on the mu2edaq cluster where I run 20 boardreaders each on mu2edaq01, mu2edaq04, mu2edaq05, mu2edaq06, mu2edaq07, mu2edaq10, mu2edaq11, mu2edaq12 (8 nodes == 160 boardreaders) along with an eventbuilder and datalogger on the same node I'm running DAQInterface on (mu2edaq11). DAQ setup script is /home/jcfree/artdaq-demo_v3_06_00/setupARTDAQDEMO.

Unfortunately, results aren't too promising. If I run using the standard develop branch, which loops over the hosts sequentially, boot time hovers around 50 seconds - e.g., 11:56:10 - 11:56:58 (today). If I try using "from multiprocessing.pool import ThreadPool" and then have the "pool" variable be an instance of ThreadPool whose argument is the # of processors on the node (56), and then run

```
pool.map(launch_procs_on_host, [host for host in launch_commands_to_run_on_host.keys()])
```

...where launch_procs_on_host is a wrapper function I've created around a block of code which is typically sequentially looped on, then it takes about a minute (e.g., 12:21:29 - 12:22:28). And to quote my own notes "Yes, I double checked that we were in direct mode, and that I had a freshly-launched DAQInterface"

As a crosscheck, I tried a different threading technique, in which I did "from threading import Thread" and used this snippet:

```
    threads = []
    for host in launch_commands_to_run_on_host:
        launch_procs_on_host(host)
        t = Thread(target=launch_procs_on_host, args=(self, host))
        threads.append(t)
        t.start()

    for t in threads:
        t.join()
```

...and again, the boot sequence took about a minute (e.g., 12:51:03 - 12:52:03).

**#2 - 12/19/2019 12:25 PM - John Freeman**

During a discussion of SBN needs at yesterday's meeting, one of those needs was "a faster boot transition". It was agreed that attempting to launch processes in a parallel across nodes was a good idea. At the time, I'd forgotten that I'd attempted this back in the summer without much success, but it's at least worth a second attempt given that this is no longer merely something the group considers a cool idea, but rather, something requested by an experiment.

**#3 - 12/19/2019 12:28 PM - John Freeman**

*- Related to Support #23764: artdaq boot sequence takes significant time for ICARUS daq added*