Instructions for 5/20/16 Allinea Workshop

When: May 20th, 1-2:30PM
Where: UNL AVH 347 and UNO PKI 250

Untar the files for the examples

The tarball containing the examples is located at /work/demo/SHARED/TrainingTools.tar.gz on Crane. To unpack them to your /work directory, run the following command:

```
Unpack the example files to /work

tar -zxvf /work/demo/SHARED/TrainingTools.tar.gz -C $WORK
```

This will create a TrainingTools folder in your /work directory with the files.

Setup the Allinea client software to use Reverse Connect

The Allinea DDT/MAP software supports a Reverse Connect feature. The GUI is installed and run locally, and information from the job running on the cluster is sent back to your laptop/workstation via SSH. This is the recommended way to use the software for interactive debugging/profiling of a job on HCC resources. Traditional X11 forwarding will also work, but is not recommended as the interface can be much slower to respond.

In order to follow along with the demos, use these instructions to setup the Allinea client on your laptop.

First, download and install the remote client software for either Windows or OS X from this page.
Alternatively, download the software directory for your OS: [OS X direct link] [Windows 64-bit direct link] [Windows 32-bit direct link]

Start the Allinea software, and choose Configure... from the Remote Launch dropdown menu.

Click the Add button on the new window.
To setup a connection to Crane, fill in the fields as follows:

- **Connection Name**: Crane
- **Host Name**: `<HCC username>@crane.unl.edu`
- **Remote Installation Directory**: `/util/opt/allinea/6.0`

It should appear similar to this:

![Remote Launch Settings](https://example.com)

Be sure to replace `demo02` with your HCC username.

Click **OK** to close this dialog, and then **Close** on **Configure Remote Connections** to return back to the main Allinea window.

Next, log in to Crane. The Allinea software uses a `.allinea` directory in your home directory to store configuration information. Since `/home` is read-only from the nodes in the cluster, the directory will be created in `/work` and symlink'd. To do so, run the following commands:

```
rm -rf $HOME/.allinea $WORK/.allinea
mkdir -p $WORK/.allinea
ln -s $WORK/.allinea $HOME/.allinea
```

### Test the Reverse Connect feature

To test the connection, choose **Crane** from the **Remote Launch** menu.

![Remote Launch](https://example.com)

A **Connect to Remote Host** dialog will appear and prompt for a password.
The login procedure is the same as for PuTTY or any other SSH program. Enter your HCC password followed by the Duo login. If the login was successful, you should see Connected to: <username>@crane.unl.edu in the lower right corner of the Allinea window.

The next step is to run a sample interactive job and test the Reverse Connect connection. Start an interactive job by running the following command.

```
Start an interactive job
srun --pty --qos=short bash
```

Once the job has started, load the allinea module and start DDT using the --connect option.

```
Start DDT
module load allinea/6.0
ddt --connect
```

On your local machine, a pop-up box should appear prompting you to accept the Reverse Connect request.

```
Choose Accept. The Remote Launch section should change to indicate you are connected via tunnel, similar to:
```

```
A new Reverse Connect request is available from 10.138.5.3:4201 for Allinea DDT.
Command Line: --connect
Do you want to accept this request?

Help       Accept       Reject
```

Choose Accept. The Remote Launch section should change to indicate you are connected via tunnel, similar to:
Once that happens, Reverse Connect is working successfully and a debugging or profiling session can be started.

During the Workshop

Compiling the examples

In order to compile and run the example programs, an MPI compiler will need to be available. For this workshop, it is suggested to use GCC 4.9 and OpenMPI 1.8. To load those modules, run

```bash
module load compiler/gcc/4.9 openmpi/1.8 allinea/6.0
```

Once the modules are loaded, you can run `make` as instructed in the various example code folders.

Starting interactive jobs for serial, OpenMP/pthreads, and MPI code

The `srun` syntax is slightly different depending on which type of code you are debugging. Use the following commands to start interactive jobs for each type.

<table>
<thead>
<tr>
<th>Code Type</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial code</td>
<td><code>srun --pty --time=2:00:00 --reservation=allinea bash</code></td>
</tr>
<tr>
<td>OpenMP/pthreads code</td>
<td><code>srun --pty --time=2:00:00 --nodes=1 --ntasks-per-node=4 --reservation=allinea bash</code></td>
</tr>
<tr>
<td>MPI code</td>
<td><code>srun --pty --time=2:00:00 --ntasks=4 --reservation=allinea bash</code></td>
</tr>
</tbody>
</table>

Please note that the `--reservation=allinea` option will only work during the workshop.