Linaro Forge Remote Client Walk-Through on Ookami

Tony <anthony.curtis@stonybrook.edu>
Linaro Forge Remote Client

- The Forge suite consists of graphical and command-line tools for debugging and profiling parallel & sequential programs
  - `perf-report`
    - A high-level overview of the behavior of your program, so you can see where to look more closely with tools such as...
  - `ddt`
    - A parallel debugger (and C / C++ static code checker)
  - `map`
    - A parallel profiler (find hotspots in code as optimization candidates)
Linaro Forge Remote Client

- **History**
  - Forge was created by a company called Allinea
    - Which was acquired by ARM
      - Which was acquired by Linaro
        - (We'll see some of these various names during usage of the tools)
  - So Forge has no dependency on the ARM compilers
    - Current version of Forge has removed the old names
      - However, the configuration directory will still be called 
        ~/.allinea for compatibility reasons

https://www.linaroforge.com/
Linaro Forge Remote Client

- Using Forge `ddt / map` directly on ookami

  ```
  ookami$ module avail linaro/*
  -------------------------------- /etc/modulefiles/aarch64 --------------------------------
  linaro/forge/22.1.4  linaro/forge/23.0
  
  ookami$ module load linaro/forge
  ookami$ forge
  ```

- Forge also has the `perf-report` command to give a high-level overview
Linaro Forge Remote Client

- Run Forge on ookami directly
  - Have to log in over Secure Shell, which ...
    - Is often really slow for a GUI
    - Requires X11 forwarding to be set up
  - Or can use Forge from the command-line
    - Useful in batch jobs

- Or ... there's a remote client that you install on your laptop / desktop machine
  - The remote client means screen updates & interaction are local
    - And thus much faster
    - The program runs on ookami, though

- Both usages integrate cleanly with SLURM
Linaro Forge Remote Client

- Follow installation instructions from
  - https://docs.linaroforge.com/23.0/html/forge/index.html
- Follow remote client instructions from
  - https://docs.linaroforge.com/23.0/html/forge/forge/connecting_to_a_remote_system/index.html

- I have used the client successfully on Mac and Linux (Fedora)
- Someone else tried it successfully under Windows too

- Examples hereafter are from a Mac, but it’s all the same really
Linaro Forge Remote Client

● Caveat
  ○ No license required for the remote client
  ○ But there are license restrictions for Forge on ookami
    ■ E.g. size of programs, number of simultaneous users
Linaro Forge Remote Client

- Using the remote client with ddt / map

Choose “configure”
Linaro Forge Remote Client

- Remote Installation Directory of Forge on ookami
  - You can find this by e.g.
    
    ```
    ookami$ module load linaro/forge
    ookami$ which forge
    /lustre/software/linaro/23.0/bin/forge
    ```

- So the location for this version is
  
  ```
  /lustre/software/linaro/23.0
  ```
Linaro Forge Remote Client

N.B. everything goes through ssh and DUO like other logins

From previous slide

See next slide
Linaro Forge Remote Client

- Remote script(s)
  - They pretend to be a login session
  - You can have any number of them
  - No fixed directory location
  - The client can have any number of configs pointing to different hosts and remote scripts
  - Next slide shows example of what worked for me
ookami$ cat ~/.allinea/remote-script
#
# set up global & personal login environments
#
source /etc/profile
source $HOME/.bash_profile

#
# at this point, it's like we just logged in to ookami
#

#
# set up extra environment for this job
#
module load fujitsu/compiler
module load linaro/forge
This example is a trivial MPI program
4 ranks, on 1 node

Go through SLURM

For Fujitsu MPI

Queue == SLURM partition, e.g. “short”

Select slurm.qtf template
Linaro Forge Remote Client
Linaro Forge Remote Client

- Configuration notes for MPI implementations
  - Fujitsu compiler / MPI
    - Choose “Open-MPI (compatibility)”
    - On command-line this is (dash dash mpi=....)
      `-mpi=openmpi-compat`
  - Any MVAPICH2
    - Choose “SLURM (generic)”