Snakemake on a Compute Cluster

General

One particularly great feature of Snakemake is that converting a workflow from a single machine to running on a compute cluster is simply a matter of activating cluster mode, and Snakemake does the rest. It may take a little experimenting to find the correct settings for your cluster but once you have these you can simply use them as the default for all workflows.

The most robust option is normally to use the **--drmaa** mechanism. This stands for "Distributed Resource Management Application API" and is available on most clusters. It allows Snakemake to have fine control over the jobs so it can submit, monitor and cancel them directly.

The Eddie system at Edinburgh

All researchers at The University of Edinburgh have access to the Eddie compute cluster.

See http://www.ecdf.ed.ac.uk for details.

This is one way to set up Snakemake for use in your personal account on Eddie. Installation via conda would also work, but uses up more disk space in your account.

1. Log in to Eddie via SSH
   - `$ ssh bobdobbs@eddie.ecdf.ed.ac.uk`
2. Make a Python3 "VirtualEnv" with at least Python 3.6
   - `$ cd /exports/applications/apps/community/roslin/python`
   - `$ ./3.6.8/bin/python3 -mvenv ~/py3.6_venv`
3. Install Snakemake and DRMAA libraries for Python3
   - `$ ~/py3.6_venv/bin/pip3 install snakemake drmaa`
4. Add a custom job runner script. Put the following into a file named `~/py3.6_venv/snakemake/jobscript.sh`:

```
#!/bin/bash -l
# properties = {properties}
sleep 2
{exec_job}
```

This uses `/bin/bash -l` instead of simply `/bin/bash` as the executor. This subtle change loads environment settings so jobs that run on the worker nodes will see the same environment as things you test on the login node. The `sleep 2` line compensates for clock skew between nodes. You can add other initialization but be careful as it's easy to break things by messing with this script.

5. Finally, add a custom wrapper script to tie it all together. Put the following into a file named `~/bin/snakemake` and make it executable with `chmod +x`

```
#!/bin/bash

# This makes DRMAA work.
if [[ "`which qsub`" =~ (.+/gridengine/.+)/bin/(.+)/qsub  ]] ; then
    DRMAA_LIBRARY_PATH="${BASH_REMATCH[1]}/lib/${BASH_REMATCH[2]}/libdrmaa.so"
    export DRMAA_LIBRARY_PATH
fi

# We need to relax the ulimits. Not sure if this is regarded as 'naughty' but
# the default limits just prevent creating any new threads and nothing works.
ulimit -s `ulimit -Hs`
ulimit -t `ulimit -Ht`
ulimit -v `ulimit -Hv`

exec ~/py3.6_venv/bin/snakemake \
    --jobscript ~/py3.6_venv/snakemake/jobscript.sh \
    --latency-wait 200 "$@"
```

You should now have the `snakemake` command available, and if you launch your workflows with the `--drmaa` flag Snakemake will submit jobs to the cluster. You will probably want to supply extra flags to `--drmaa` to, eg., set the queue name and logfile locations.

Other Compute Clusters

If you have access to an institutional compute cluster with a shared filesystem architecture then it is almost certain that Snakemake can be set up to run on it. We'll be happy to help you with the setup process and point you to useful resources, so please get in touch with the course tutors directly if you are struggling.

Commercial Cloud Computing

The main author of Snakemake has put a lot of work into making Snakemake operate in cloud computing environments, where access to compute resources is highly abstracted via systems such as Kubernetes and shared file systems are replaced by object storage. However, compared to the relatively simple process of moving from a single system to a compute cluster, the step to cloud involves more effort both to set things up and to make effective use of the cloud resources.

At the time of writing, the course tutors do not have direct experience of setting up workflows this way.

If you want to try yourself, see the Snakemake documentation:

> https://snakemake.readthedocs.io/en/stable/executable.html#cloud-support

And also look out for useful blog entries like for example:

> https://blog.liang2.tw/posts/2017/08/snakemake-google-cloud/

Some other useful resources

Snakemake questions on Biostars.org

> https://www.biostars.org/t/snakemake/

Sequana project - curated workflows using Snakemake

> https://sequana.readthedocs.io/en/master/

Cooking Moussaka with Snakemake (!)

> https://github.com/deepalivasoya/Snakemake-food