Xgrid

What is Xgrid?

Xgrid is proprietary software from Apple which allows an ad hoc group of Macs (a collection of desktop machines, servers, old machines) to work as a distributed computing platform. The system works by allowing idle machines (known as "agents") to donate their resources to computationally intensive tasks, while they are not being used by their owners. The collection and distribution of tasks is handled by a central machine, known as a "controller". Every copy of the latest version of OS X ("Tiger") comes with a copy of the xgrid package.

What can I use Xgrid for?

The most straightforward use of Xgrid is for tasks which are computationally intensive but can easily be split into multiple sub-tasks. This can often mean multiple runs of the same program with different inputs each time. A good example would be multiple BLAST jobs, or multiple runs of a simulation program with different parameter sets.

How can I add my (OS X 10.4 – Tiger) Mac to the Xgrid?

To add your machine to the Xgrid :

1. Start up "System Preferences" and click "Sharing"
2. Select "Xgrid" in the services pane on the left and click "configure" on the right
3. Under "Use a specific controller" enter the controller name. This is the DNS name of the machine which handles the distribution of Xgrid tasks. Currently this is "dirofilaria.cap.ed.ac.uk"
4. Select whether you want your machine to behave as a dedicated or part-time agent by selecting "agent accepts all tasks all the time" or "agent accepts tasks only when idle", respectively.
5. Set the Authentication Method to "none".
6. Click on "OK" and then "Start" to add your machine to the Xgrid. You can remove your machine from the Xgrid at any time by clicking "Stop".
7. You may also want to set your machine's "sleep" settings to "never" if you always want it to be available when idle.

How can I send my BLAST jobs/simulation program/... to the Xgrid?

Xgrid jobs are sent from a terminal command line in OS X. In order to use the Xgrid you need to obtain the "client" password. The Xgrid command line syntax is as follows:

```
xgrid -h dirofilaria.cap.ed.ac.uk -p <client_password> -job
<(submit|results|attributes)>
```

Following the "-job" the user can tell xgrid that they want to send ("submit") a job, find out the current status of a job ("attributes") or retrieve the results ("results") of a job they have submitted. When a task is submitted, it is tagged with a unique identifier which can be used to query, retrieve and delete jobs.

Sending directories

If the job does not use a program which is likely to be available on every agent in the cluster, in exactly the same location (something like /bin/echo for example), then the program, and whatever input files it requires **all** need to be submitted to the Xgrid. This can be done using the "-in" switch, which specifies a directory, the entire contents of which will be sent to every machine on which the job runs. Multiple directories can be submitted in this way. **NB** The path to the submitted directories ***changes*** when they are sent to the xgrid. For example:

If I'm working in /Users/dgaffney/blast, and I want to submit a shell script foo.sh which is located in /Users/dgaffney/blast/bin directory this can be done as follows

```
xgrid -h dirofilaria.cap.ed.ac.uk -p <client_password> -job submit
Users/dgaffney/blast/bin/foo.sh -in bin/
```

Notice that the path to the shell script is "Users/dgaffney/blast/bin/foo.sh" **not** "/Users/dgaffney/blast/bin/foo.sh". The former is the path to the script on the agent machine (which you need to specify in order for the job to actually run on the agent), the latter is the path on the machine on which the job was submitted (the client machine).

Retrieving results

When jobs are submitted to the Xgrid, they are tagged with an ID number. This ID number is subsequently used to query and retrieve the results of the job. To retrieve the results of job number 36 we would enter:

```
xgrid -h dirofilaria.cap.ed.ac.uk -p <client_password> -job
results -id 36 -out ./
```

Note that we need to explicitly state that the results/output files (if any) are to be dumped into a certain directory. This is done using the "-out" flag (here results are dumped into the current working directory).

xgrid_blast.pl

A very basic script (xgrid_blast.pl) has been written to allow you to submit BLAST jobs to the xgrid. For this script to work you need

1. A local installation of NCBI blast. The path to this directory needs to be specified in the xgrid_blast.pl source (variable $path_to_blast_directory)
2. A subdirectory in your current working directory containing one or more query files, in FASTA format, ending in '.fa' or '.fsa'
3. A subdirectory in your current working directory containing the BLAST databases you wish to BLAST against
4. ***No*** directory or file in your current working directory named "data" (uppercase/lowercase/mixture) - this is required for BLAST matrices.

xgrid_blast.pl writes a number of simple shell scripts which can be submitted at run time or later. Hopefully, these shell scripts will give you an idea of the syntax required to submit BLAST jobs to the xgrid. An example run of xgrid_blast.pl is as follows. I have in my current working directory a subdirectory named "queries" which contains a number of nucleotide sequence query files (ending in .fa or .fsa) which I want to blast against the database test_db in the subdirectory "databases". I would do this as follows:

```
xgrid_blast.pl --queries /Users/dgaffney/blast/queries/ --database
/Users/dgaffney/blast/databases/test_db -e 1e-10 --program blastn
-p f
```

The "--queries" and "--database" switches give the path to these directories on the client machine, "--program" specifies which blast program is to be used, "-e" denotes the threshold e-value for the queries and "-p" denotes whether the query/database files are protein sequences (value "t") or nucleotide (value "f").