Piraña: PCluster

Installation guide and Manual

June 29, 2011

Contents

Chapter 1

PCluster

The PCluster system is aimed to be a system for distributed NONMEM computing for smaller modeling groups, e.g. in hospital or academic settings. If budget poses no problems, other cluster solutions are probably more adequate. Our major development goal was to develop a system that could be installed on an existing (Windows-based) network, and that would use spare CPU cycles of non-dedicated PCs in the network.

1.1 Description and requirements

The infrastructure uses PC's in a standard Windows network environment, which can e.g. be PC's dedicated to running NONMEM, or PC's of co-workers, or a combination of both. The PCluster can be set up without the need for vast hardware/software knowledge. Distribution to multi-core CPU's is possible, and tested up to a total of 40 CPUs.

The infrastructure requires a shared network-drive accessible by all clients (standard desktop PC's in a network environment). On this shared network drive, a version of Perl and Fortran must be present. Furthermore, the PCluster *daemon* Perl-script needs to run in the background on each node. On the client computer, i.e. the modeler's computer, Piraña needs to be installed. Below, more information is provided on how to set this up.

1.2 Installation

First, it is necessary to reserve up some harddisk space on your network, to be accessible (read and write) by all clients, mounted as a drive on all nodes (let's assume this to be U:). Copy the contents of the folder `install_on_each_node` in the `/cluster` folder in the Piraña home folder to this drive. Next, the daemon script (or the compiled version of it) should be installed and running on each client in the network. For this, copy the folder \pcluster\install_on_each_node located in the Pirana home folder in the harddisk root (C:\). The daemon can be started manually by executing the Perl script daemon.pl or the daemon.exe. Before starting, check the settings in C:\pcluster\pcluster.ini. This file contains the info for the client, e.g. the number of CPUs to use, and how to connect to the shared drive. Instead of starting the daemon manually, it is recommend to run the daemon as Windows service, which has the advantage that users are able to log in and out without compromising the integrity of the PCluster, and that the service is automatically started when the PC is started. To install this service, run the Perl script inst_daemon.pl:

```
perl inst_daemon.pl
```

This will create the information in the Windows registry for the PDaemon service. If no error is reported, the service can now be started from the command line with:

```
NET START PDaemon
```

It has been reported that the daemon sometimes isn't working properly (i.e. not requesting PsN/nfme runs), while the script ís working properly when it is run manually. Running the daemon manually is however a problem, as a console window will appear each time a run is processing, which is annoying to the principal user of the machine. A workaround for this is to download a tool called CMDOW (available from http://www.commandline.co.uk/cmdow/). If you start the following command (e.g. in a batch script at startup), no consoles will appear:

```
u:\cmdow.exe /run /hid u:\perl\bin\perl.exe c:\pcluster\pdaemon.pl
```

Note: Thanks to Andreas Lindauer for suggesting this workaround.

Perl and PsN

Make sure Perl (with PsN installed) is available on the cluster-drive (e.g. in U:\Perl). Also check that you have the following Perl modules installed: LockFile, Sys::Hostname and File. Since this option hasn't been fully developed yet in PsN, it is necessary to make a few changes in some PsN source files (using PsN 3.1.0 as a template).

- In nonmem.pm in the PsN directory, add the following lines at the top of the source-file:

  ```
  my $fortran_dir = 'U:\MinGW\bin';
  unless ($ENV{'PATH'} =~ m/$fortran_dir/) {
    $ENV{'PATH'} = $fortran_dir.";".$ENV{'PATH'};
  }
  ```

This ensures the Fortran compiler is in the path (change U:\MinGW\bin' to the fortran path that you use). Of course this is not necessary if this directory is already permanently in the PATH environment variable.

- In tool\modelfit.pm change line 3014:

  ```
  require File::Temp; # qw/tempfile tempdir/;
  require Sys::Hostname;
  require LockFile::Simple; # qw/lock unlock trylock/; #Non-standard module
  ```

  into

  ```
  use File::Temp qw/tempfile tempdir/;
  use Sys::Hostname;
  use LockFile::Simple qw/lock unlock trylock/;  # Non-standard module
  use File::Basename;
  ```

  In the same file, move the contents of line 3063:

  ```
  return $jobId;
  ```

  to line 3057, just after the line with 'unlock $jobId;'.

  In the same file, around line 3105, change:

  ```
  if( -e "$ZinkDoneDir$jobId" ){
  ```

  into

  ```
  use File::Basename;
  my $job = basename($jobId);
  if( -e "$ZinkDoneDir/$job" ){
  ```

- In psn.conf file in the PsN folder on the cluster, edit the following lines with your settings (also remove leading semi-colons):

  ```
  40 : perl = U:\Perl\bin\perl.exe
  82 : zink_dir = U:
  105: default=U:\nmvi,6
  ```

  Of course, the other PsN settings should be set correctly as well.

Fortran (g77)

If you want to use the PsN-toolkit on the PCluster, you must provide access to a Fortran compiler for each client in the cluster. This is because PsN runs are compiled after distribution to a client. If you only make use of Pirana's own distribution capabilities, only a Fortran compiler on the modeler's own computer is required. The recommended way is to use a central installation of Fortran on the cluster drive, e.g. g77 supplied with MinGW. For this, e.g. install MinGW with g77 on your local computer. After installation, copy the folder MinGW to the cluster drive.

Pirana preferences

In Piraña the following preferences should be updated:

zink_host: e.g. ZinkHost, A host name for the Zink / PCluster, should correspond with the hostname specified in psn.conf.

And in 'software settings', change:

f77_dir: Path to Fortran on the cluster-drive e.g. U:/MinGW/bin

perl_dir: Path to Perl on the cluster-drive e.g. U:/Perl

psn_dir: Path to PsN on the cluster-drive e.g. U:/Perl/site/lib/PsN_2_3_1

Troubleshooting

The PCluster, although it functions quite adequately in the current setup, is actually still in beta phase. If you have questions regarding the set-up or functionality of the PCluster, please send a mail to the mailinglist (info@pirana-software.com) or contact the Piraña development team directly.

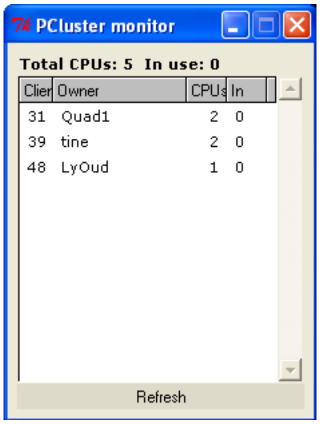
1.3 Working with the PCluster

Using the PsN toolkit on PCluster

This functionality is enabled by specifying '-run_on_zink' on the PsN command line from the Piraña window, after selecting a PsN tool from the toolkit. You should leave the PsN console window open until the run is finished, since the instance of PsN that is running locally will take care of the run distribution and calculation and formatting of data files after finishing the cluster run(s).

Cluster monitor

Using the cluster monitor ('View → Cluster monitor') the activity and availability of clients can be monitored. Note that status of the client might take a half a minute to refresh with the up-to-date information, due to the interval used by the daemon-script.



Total CPUs: 5 In use: 0

Clier	Owner	CPUs	In
31	Quad1	2	0
39	tine	2	0
48	LyOud	1	0

Figure 1.1: PCluster monitor