Subject: To PVM or not to PVM
Posted by Robert Barnett on Tue, 07 Dec 2004 00:53:13 GMT
View Forum Message <> Reply to Message

Hi,

This post is about Parallel Virtual Machine (PVM). PVM is useful for creating distributed computing clusters. Using clusters a high computational load can be split up amongst many processors on a network. More information about PVM is avilable from http://www.csm.ornl.gov/pvm/pvm_home.html.

PVM comes with a C library. However, the closest thing I have found to a IDL dynamically loadable module (DLM) is at http://www.kilvarock.com/software/idltopvm.htm. PVM is completely different to IDLs multi-threading. In fact it has been argued that IDL is not suitable for clustered applications. Details about IDL and Multithreading are outlined at http://www.rsinc.com/services/techtip.asp?ttid=3252

I've written a distributed computing application which just plays musical chairs with sav-files and raw-data files. I've been meaning to use PVM for some time, but it doesn't seem like it's worth the effort.

Some of the problems with PVM be appear to be:

* I have to compile and set up PVM independantly for each platform.
* PVM requires rsh or ssh. Neither are built-in services provided by windows workstations
* There is no published DLM wrapper for the PVM C library.

I don't know a great deal about distributed computing. Most people cringe at my solution.

My solution is based on a master/slave model, where the master creates jobs for the slaves to do.

* I have a script which accepts socket connections from any slave/master. This script manages how jobs are received from the master and distributed amongst the slaves. It does not actually transfer data. The script is written in a language which is able to handle multiple threads (currently python but there is also a perl version).
* The actual data is sent between each machine using FTP (passive connections). The master has a FTP server with files stored on the ramdisk.

There is maximum overhead of 100ms (on client) per interprocess signal. Most of that overhead appears to be opening a new socket and sending the (small amount of) data via passive FTP. Given the nature of my project,

this overhead is not of much concern (about 10% of total computation).
To make the solution completely cross platform, I went for FTP rather
than a NFS or Samba drive.

Are there any reasons why I should be using PVM? Has anyone has notable
success in using PVM and IDL in an inhomogenous network? Is this a good
method for setting up a prototype distributed computing application?
Would anyone be interested in my adapting the code for public release?

Cheers,

--

nrb@
Robbie Barnett
imag
Research Assistant
wsahs
Nuclear Medicine & Ultrasound
nsw
Westmead Hospital
gov
Sydney Australia
au
+61 2 9845 7223