

ParallelKnoppix Tutorial

Michael Creel*

7th October 2005

Abstract

This note shows how to set up a non-dedicated HPC Linux cluster for MPI- and/or PVM-based parallel processing using ParallelKnoppix, a bootable CD. The computers that form the cluster are left in their original state once the cluster is shut down. The setup procedure, which takes only 5-10 minutes to perform, is explained in detail. Next I give some brief examples of use of the cluster to execute parallel programs. Finally I discuss the simple steps that can be taken to personalize the CD to add programs and data.

Keywords: parallel computing; HPC cluster

JEL codes: C63; C87; C88

1 Introduction

ParallelKnoppix¹ (P-KPX) is a bootable CD that allows creation of a high performance computing (HPC) Linux cluster in very little time². The cluster is temporary - it can be created using computers that have their entire hard disk(s) occupied by another operating system and users' data, and this previously existing installation will be intact and unchanged once the cluster is shut down. P-KPX has been used to set up clusters of up to 50 computers, and it is configured to allow creation of clusters of up to 200 computers. The results presented in Creel (in press) and Creel (2005) are illustrative of the computational speedups that can be obtained from parallel computing applied to problems from econometrics and real business cycle macroeconomic modeling. Those results were obtained using P-KPX in university computer rooms that are ordinarily used for teaching and students' work, but which are closed at night. This tutorial shows how to create and use a HPC cluster, step-by-step, using screenshots. It also explains the simple steps that can be taken to personalize the P-KPX CD to add programs and data. The latest version of this tutorial is always available at <http://pareto.uab.es/wp/2004/62604.pdf>.

2 Prerequisites

P-KPX is designed to work with computers of the IA-32 architecture, which includes Intel Pentium IV and Xeon processors, as well as the AMD Sempron and Athlon processors. Future versions will support the IA-64 architecture when it is more widely adopted. The main requirement to guarantee a simple setup process is that the network cards of all slave nodes in the cluster allow booting across the network using PXE. All newer network cards support this option, though it may be necessary to configure the BIOS of the slave nodes to enable this feature. Since the slave nodes will obtain their IP addresses from the dhcp server running on the master node, it may be necessary to isolate the slave nodes from any other dhcp server that is running.

*Department of Economics and Economic History, Edifici B, Universitat Autònoma de Barcelona, 08193 Bellaterra (Barcelona) Spain. *email:* michael.creel@uab.es; *Tel.:* 93-581-1696; *FAX:* 93-581-2012. This research was supported by grants SGR2001-0164 and SEC2003-05112.

¹<http://pareto.uab.es/mcreel/ParallelKnoppix>

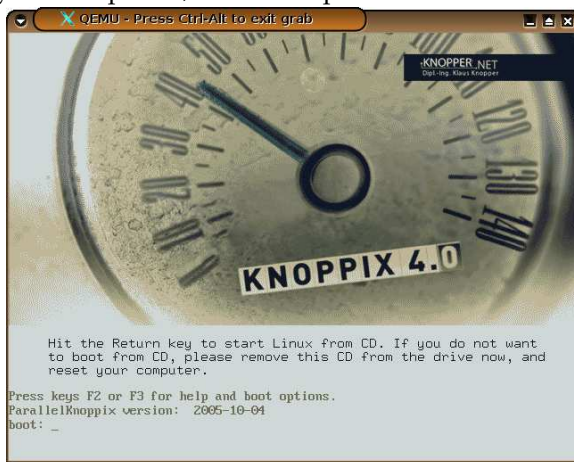
²An experienced user can set up a cluster in around 5-10 minutes, independently of the size of the cluster.

3 Setup

This section discusses the initial setup of the cluster, up to the point where parallel programs may be executed.

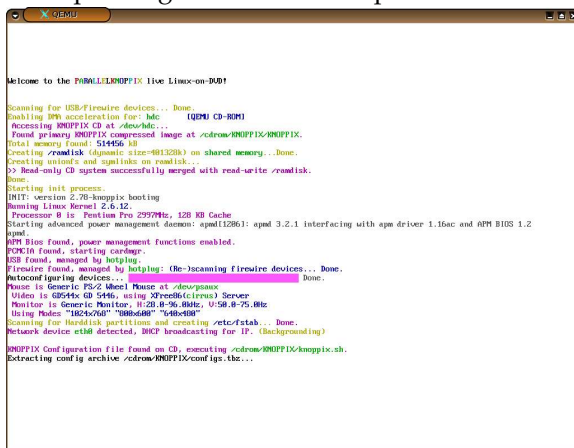
3.1 Booting up

The master computer must be booted using the P-KPX CD. You may need to enter your BIOS setup routine to configure the computer to boot from CD. Once this is done, place the CD in one of your computers, and boot up. You will see something similar to the following:



This image clearly reflects the fact that P-KPX is based upon the Knoppix³ Linux distribution. Note that you can press F2 and/or F3 to see information about boot options. Also, the version number of P-KPX is indicated.

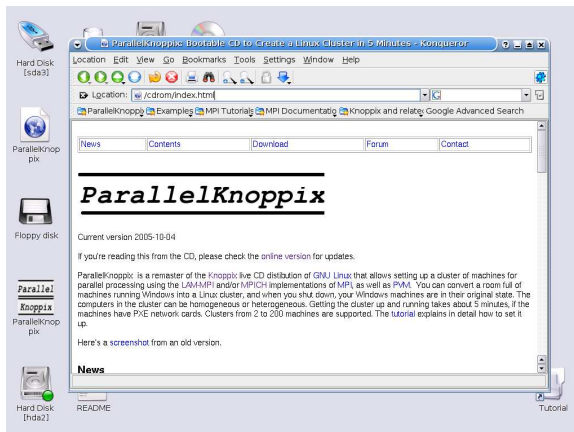
After pressing <ENTER> the computer starts to boot from the CD. After a bit we see



P-KPX uses Knoppix's excellent hardware detection to automatically configure the mouse, video card and network card. This same autodetection will be used for all the computers in the cluster, which greatly simplifies the creation of a cluster using heterogeneous computers.

When the master computer has booted we are in the KDE desktop environment:

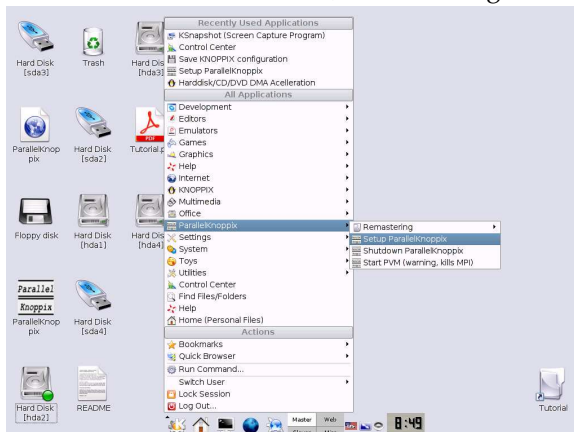
³<http://www.knopper.net/knoppix/index-en.html>



This is an intuitive graphical user environment. A single left click is used to choose options, open files, etc. A right click usually reveals options and information. When in doubt, right click.

3.2 Configuration

All configuration is done using a script that you can start from the ParallelKnoppix/SetupParallelKnoppix menu item. To access the menu, click on the gear icon to the left of the bottom panel:



The script configures the first network card (the one that is given the name `eth0`) to use the IP address 192.168.0.1. If you have more than one networking card, you may need to switch cables so that `eth0` is the card which connects to the cluster. Next, the terminal server is started to boot your slave computers. First you get some information:



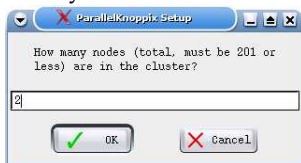
Click on OK.

Next you are asked if you want to configure the terminal server:



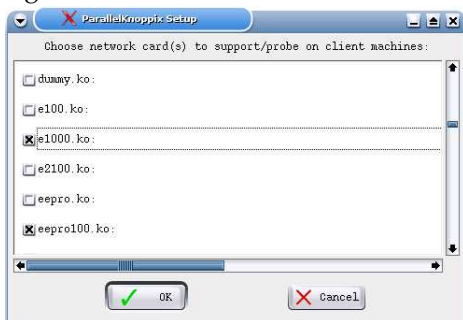
Click OK.

Next you are asked how many nodes are in your cluster:



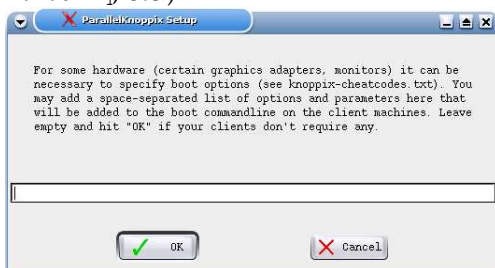
Enter the number, including the master node you're sitting in front of.

Next, you need to choose the network card types that are in your cluster, so that the initial kernel used to perform the PXE boot is able to set up networking on the slave nodes. There is a dialog to do this:



Select ALL network card types in the cluster, then click OK.

Next, you may need to pass some special options to get your nodes to boot (such as `acpi=off`, `pci=biosirq`, etc.).⁴

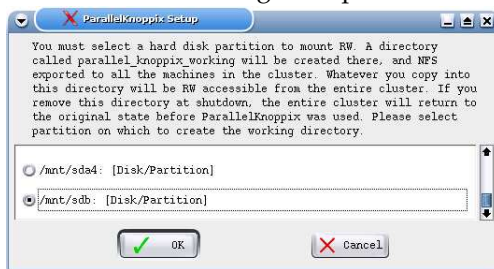


I just click OK in this example, since I know that the slave node does not require options.

Now, you need to select a partition on the master node's hard drive on which to create a working directory. This directory will be made writable, and NFS exported to all machines in the cluster. Any data or programs placed in this directory will be available to all the nodes. The working directory will be named "parallel_knoppix_working" to minimize the chances that an existing directory has the same name. The script will only let you use partition types that are safe to write to, which includes all common partition types except NTFS. If the hard disk is entirely occupied by NTFS partitions (not an uncommon case when Microsoft Windows (R) is

⁴ Use of these options is explained at http://www.knoppix.net/wiki/Cheat_Codes.

pre-installed on the computer), one can use a USB flash memory for the working space. This is what I do in the following example:

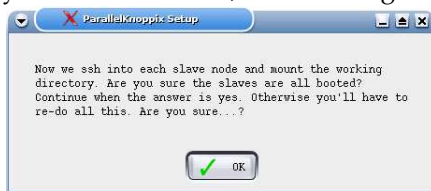


The following message appears confirming that the partition has been mounted:

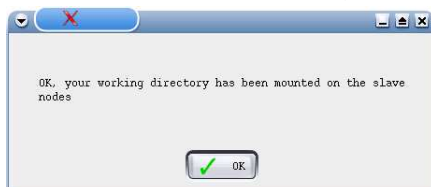


A link is created on the desktop that points to the working directory, to make navigation comfortable.

Next, the working directory must be mounted on the slave nodes. This is only possible after they have been booted, so a warning message appears:



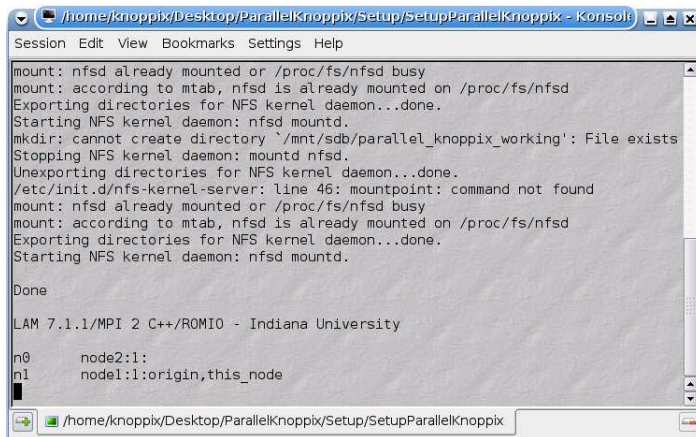
Once the slave nodes are booted, click OK, and you will receive a confirmation:



Next, the cluster is `lambbooted` so that LAM/MPI⁵ may be used to run MPI-based parallel programs.⁶ We can see that the 2-node cluster of this example is successfully `lambbooted`:

⁵<http://www.lam-mpi.org/>

⁶LAM/MPI is a library of functions that implement the MPI standard (<http://www-unix.mcs.anl.gov/mpi/>). An alternative library, MPICH (<http://www-unix.mcs.anl.gov/mpi/mpich/>) is also included on the P-KPX CD, but its use is not discussed here.



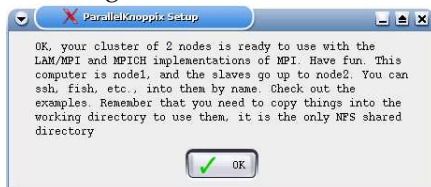
```
mount: nfsd already mounted or /proc/fs/nfsd busy
mount: according to mtab, nfsd is already mounted on /proc/fs/nfsd
Exporting directories for NFS kernel daemon...done.
Starting NFS kernel daemon: nfsd mountd.
mkdir: cannot create directory '/mnt/sdb/parallel_knoppix_working': File exists
Stopping NFS kernel daemon: mountd nfsd.
Unexporting directories for NFS kernel daemon...done.
/etc/init.d/nfs-kernel-server: line 46: mountpoint: command not found
mount: nfsd already mounted or /proc/fs/nfsd busy
mount: according to mtab, nfsd is already mounted on /proc/fs/nfsd
Exporting directories for NFS kernel daemon...done.
Starting NFS kernel daemon: nfsd mountd.

Done

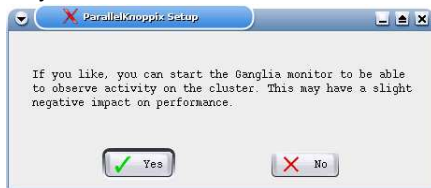
LAM 7.1.1/MPI 2 C++/ROMIO - Indiana University

n0      node2:1:
n1      node1:1:origin,this_node
```

A message informs that the cluster was successfully created:



You are given a chance to start the ganglia monitoring daemon, to be able to observe the activity on the cluster:



If you click YES, you will see the following information:

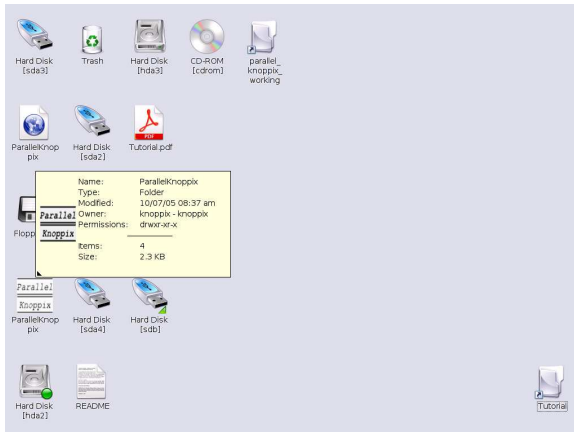


If you go to the address in the previous figure, you will see information about the cluster (an example appears below). There is a bookmark stored in the two web browsers included on the CD.

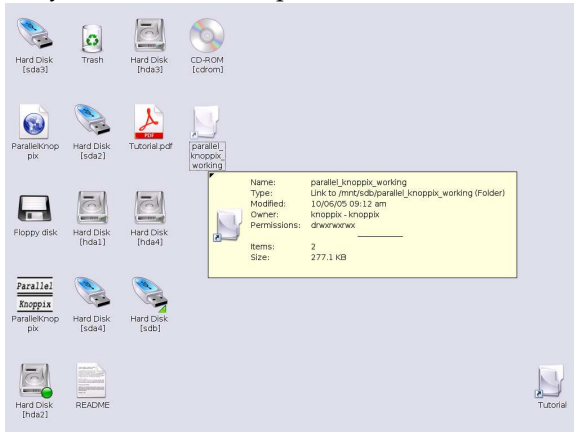
The cluster is now ready for use. The master node has the IP address 192.168.0.1. The slaves have IP addresses that go up to 192.168.0.x, where x is the total number of computers in the cluster. Each computer may be accessed by its hostname "nodex". Inside the cluster, you can become the root user on any node simply by entering `su` in a terminal, or by executing a command preceded by `sudo`. To gain root access from outside the cluster, the password is "parallelknoppix". This is clearly insecure, and its implications are discussed below.

4 Use

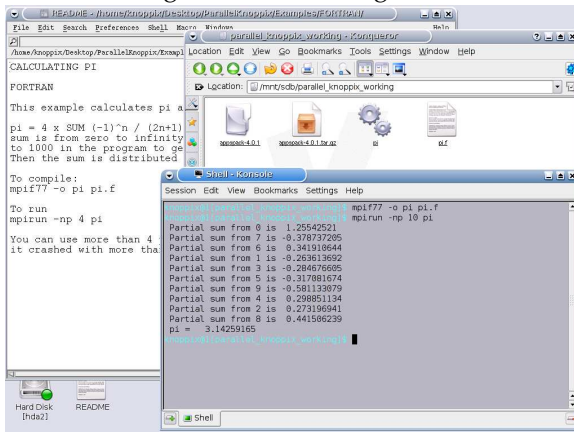
To run programs in parallel, they must be put into the working directory. There are a number of examples in the ParallelKnoppix directory on the Desktop:



They will need to be copied to the `parallel_knoppix_working` directory:

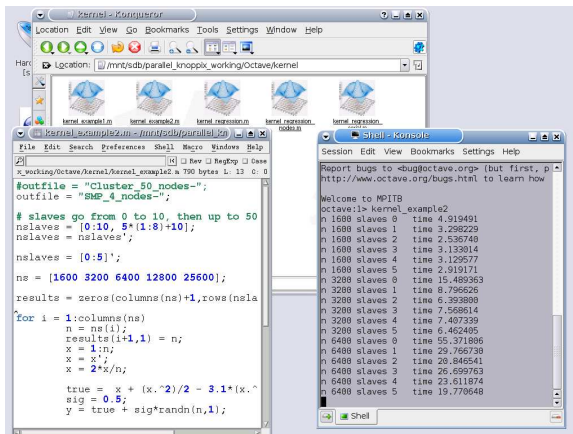


In the following figure, we see that the `pi.f` example has been copied from the `./FORTRAN` subdirectory of `./Desktop/ParallelKnoppix/Examples` to the working directory, and compiled and run following the instructions given in the README file that accompanies the example:

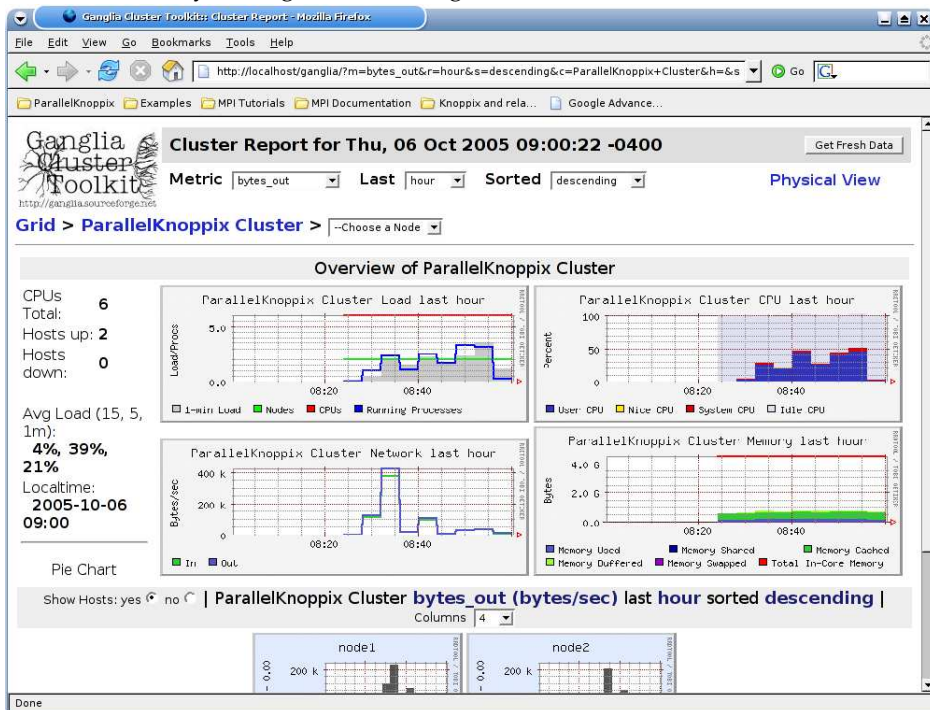


Ten software processes were run on the 2 computing nodes, 5 processes per node.

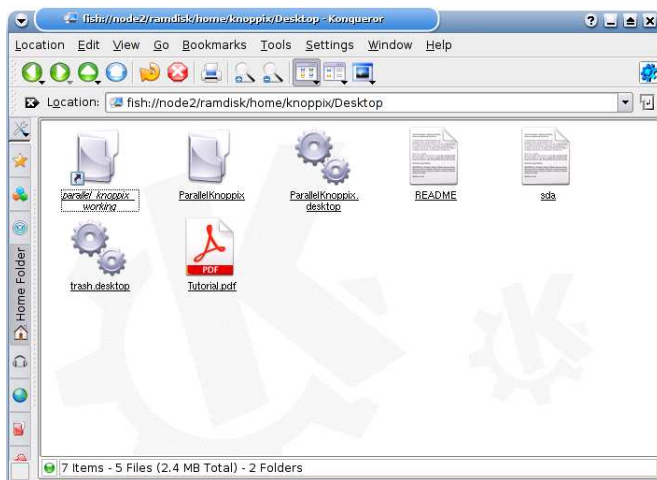
A more interesting example is kernel regression, in the `./Desktop/ParallelKnoppix/Examples/Octave` directory. Once this entire directory is copied to the working directory, it may be run by opening a terminal in the `./Octave/kernel` subdirectory inside the working directory (open a terminal by pressing F4), typing `octave` to start octave, and then typing `kernel_example2`:



The previous figure shows the output obtained on a cluster made up of a uniprocessor 3.00 GHz Pentium IV master node using hyperthreading, and a 2x3.06 GHz Xeon SMP slave node, also using hyperthreading. So there are 3 real CPUs and 6 virtual CPUs. The cluster was lambooted to send 2 of every 3 processes to the slave node. Such nonstandard environments are easily created by editing the /home/knoppix/hostdef file, issuing the lamhalt command, and then lamboot hostdef to start the new configuration. In the previous figure we observe that most improvement is obtained by going from 0 slaves to 2 slaves (so the 3 real processors are all being used), but that 5 slaves is better than 2 slaves (so that all 6 virtual processors are used). We observe that hyper-threading leads to a modest performance improvement. Following we see the ganglia monitor's record of the activity during the kernel regression run:

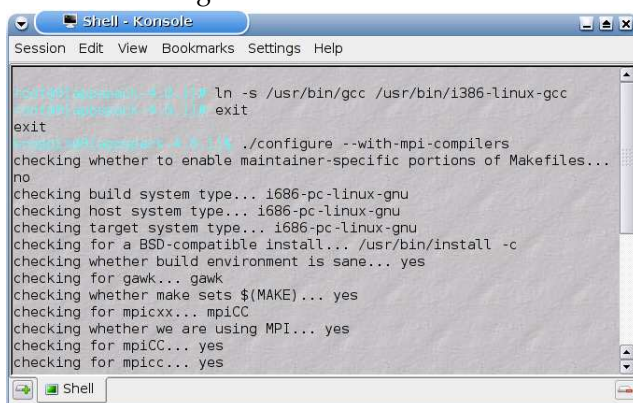


We may need to copy files from other computers onto the cluster. This can easily be done using the fish:// protocol, if we use the konqueror web browser:

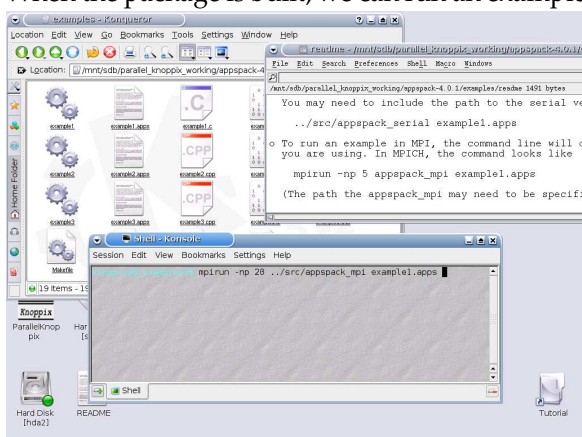


You can use this to copy files to and from any machine that you can ssh into. This is a convenient means of getting needed information onto the cluster.

One may be interested in using packages that are not included on the P-KPX CD. An example is APPSPACK⁷, a program for derivative-free solution of nonlinear programs. One can download and decompress the source code into the working directory. To compile, we need to do a trick to set things up as is expected by the package. This trick, as well as the configuration, is seen in the following:



When the package is built, we can run an example by following the information in the README:



This gives the results:

⁷<http://software.sandia.gov/appspack/>

```

Session Edit View Bookmarks Settings Help

Final Min: f= 3.125e-04 x=[ 0.000e+00 -1.250e-02 ] step=3.125e-02 tag=64 state
=Evaluated Success: 53

Final Directions:
0 : d = [ 2.000e+00 0.000e+00 ] step = 7.812e-03
1 : d = [ -2.000e+00 0.000e+00 ] step = 7.812e-03
2 : d = [ 0.000e+00 2.000e+00 ] step = 7.812e-03
3 : d = [ 0.000e+00 -2.000e+00 ] step = 7.812e-03

Number of Cached Function Evaluations: 12
Number of Evaluations: 60

Evaluation Breakdown by Message Type:
Success: 60

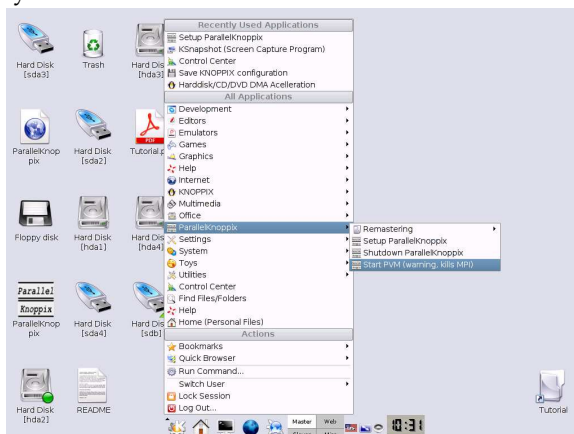
Evaluation Breakdown by Processor and Message Type:
Worker #1
Success: 11
Worker #2
Success: 9
Worker #3
Success: 5
Worker #4
Success: 19
Worker #5
Success: 11
Worker #6
Success: 3
Worker #7
Success: 1
Worker #8
Success: 1

knoppix@knoppix:~$

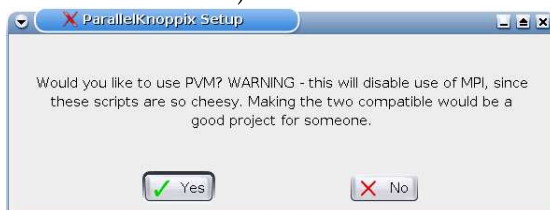
```

Thus, we see that as long as packages are built in `parallel_knoppix_working`, they will run on the cluster. Once a package has been built, a means of saving it for use in future sessions is discussed in the next section.

P-KPX also includes PVM⁸ for executing parallel programs. To start PVM, there is a menu entry:



PVM and MPI can't currently be used at the same time. There is a warning (and a veiled invitation to contribute):

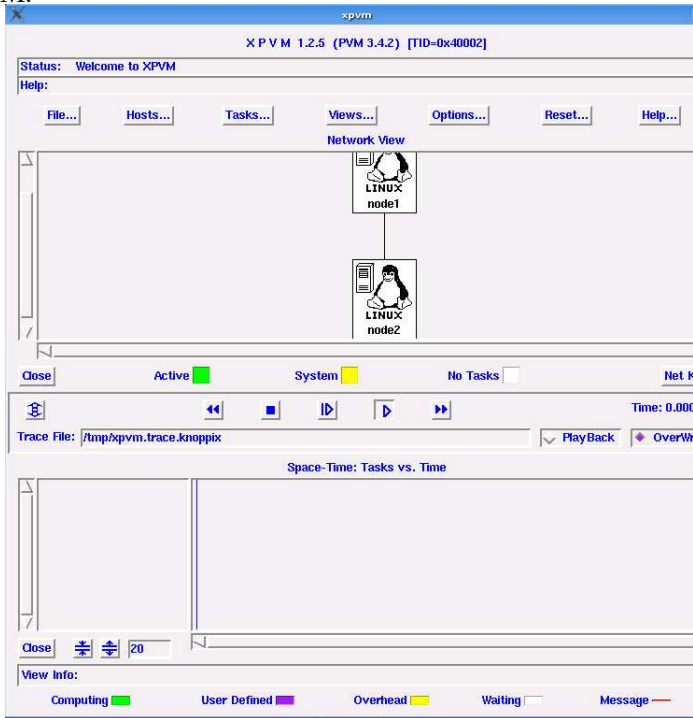


If you click OK you see the following:

⁸http://www.csm.ornl.gov/pvm/pvm_home.html



If you issue the command `xpvmm`, you will see that your nodes have been integrated into the PVM:



4.1 Saving your work for another session

You should place your personal files in `/home/knoppix/Desktop`, since that directory is easy to save and restore. To save files that are within the `/home/knoppix/Desktop` directory, I recommend the following.

1. Use the menu item `Knoppix-> Configure-> Save KNOPPIX configuration`. Select the **first and second options only** (Personal configuration and All data in Desktop).
2. Then save to one of the disk partitions that you are offered. I recommend using a USB key drive for portability reasons.

Remember that the `parallel_knoppix_working` icon on your Desktop is not really a directory on the Desktop, it is a link to a directory on a partition on some storage media. To save work in the working directory, it must be copied to a real folder on the Desktop.

To re-load your work, do the following.

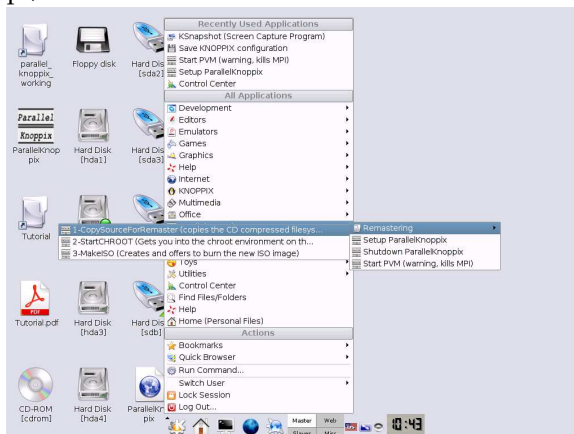
1. Then re-start ParallelKnoppix when you like.
2. When you are in KDE, mount the partition where you saved your work.
3. Open a terminal there, and type `tar -jpPxf configs.tbz`

Now whatever files were in `./Desktop` are restored. This provides a relatively easy way of adding your own files and data for multi-session use.

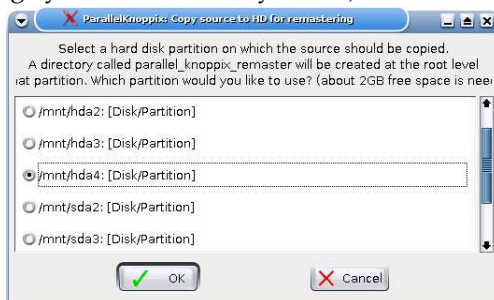
5 Personalization (Remastering)

It may be that you need to add programs to the CD. This section shows how you can do it. I assume you already have networking configured so that you can access the Internet, if not, see the KNOPPIX->Network/Internet->Network card configuration entry in the KDE menu.

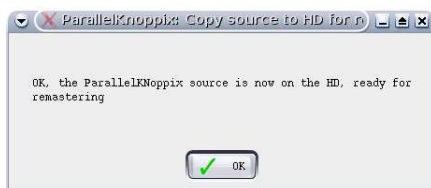
First, copy the source to the hard disk. Start the script using the 1-CopySourceForRemaster script, in the menu



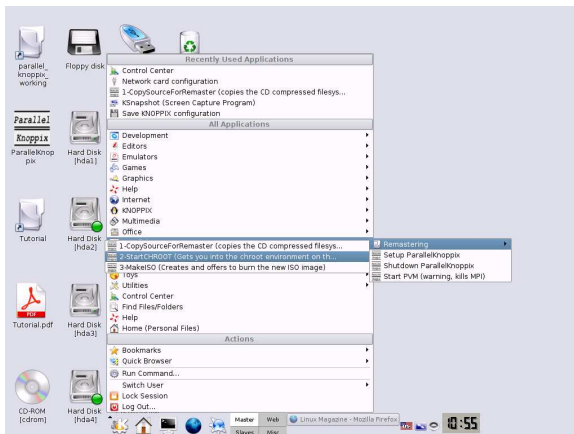
This will ask you where you would like to copy the source to. Select a disk partition with enough space (4GB should be enough - you need room for the decompressed source and for the ISO image you will eventually create):



After you choose a partition and click OK, you will need to wait a while while the source is copied from the CD to the hard drive. Eventually you'll see a message telling you it worked:



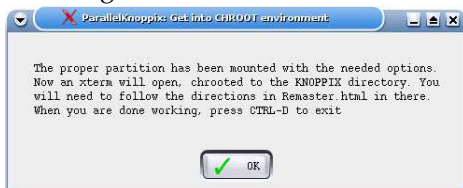
Next, run the 2-StartCHROOT script:



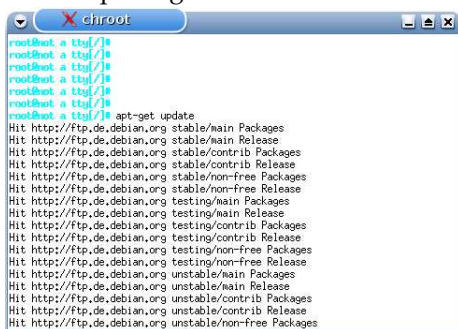
You are asked to where you copied the source in the previous step:



You are given some information:



Then your `chroot`'ed window opens up. Note the pleasing shade of blue that reminds you that this is not your ordinary environment. Do an `apt-get update` to get information about the latest versions of packages:



... then install your favorite missing package using `apt-get install`:

```

chroot
Hit http://security.debian.org testing/updates/contrib Release
Hit http://security.debian.org testing/updates/non-free Packages
Hit http://security.debian.org testing/updates/non-free Release
Ign http://ndiswrapper.sourceforge.net ./ Release
Hit http://kanotix.com ./ Packages
Hit http://kanotix.com ./ Release
Reading Package Lists... Done
root@chroot:~# apt-get install frozen-bubble
Reading Package Lists... Done
Building Dependency Tree... Done
The following NEW packages will be installed:
  frozen-bubble
0 upgraded, 1 newly installed, 0 to remove and 436 not upgraded.
Need to get 74.9kB of archives.
After unpacking 285kB of additional disk space will be used.
Get:1 http://ftp.de.debian.org testing/main frozen-bubble 1.0.0-6 [74.9kB]
Fetched 74.9kB in 0s (173kB/s)
Selecting previously deselected package frozen-bubble.
(Reading database ... 87781 files and directories currently installed.)
Unpacking frozen-bubble (from .../frozen-bubble_1.0.0-6_1386.deb) ...
Setting up frozen-bubble (1.0.0-6) ...

root@chroot:~#

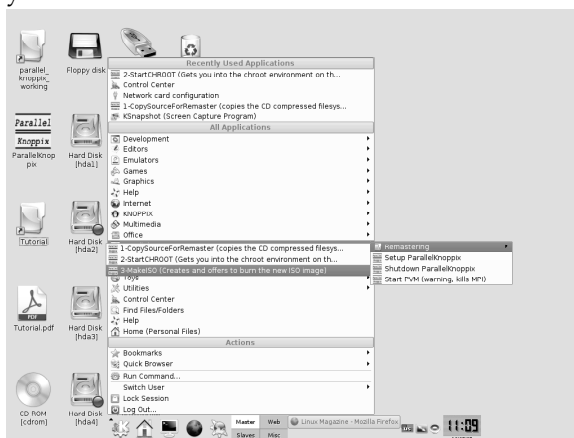
```

You don't want a lot of .deb files in your /var/cache/apt subdirectory making your CD image become enormous, so before exiting, issue the apt-get clean and apt-get autoclean commands. Also, you'll generally want /etc/resolv.conf empty, so issue the command echo "" > /etc/resolv.conf

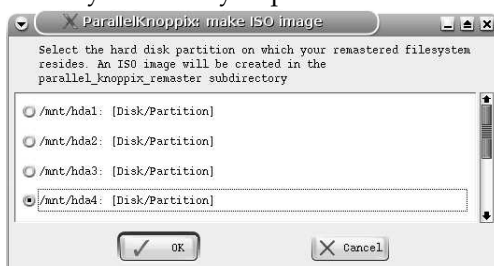
When you are done, type CTRL-D to exit the CHROOT environment, and you see:



Next, you need to make the image of your modified P-KPX. The 3-MakeISO entry will do this for you:



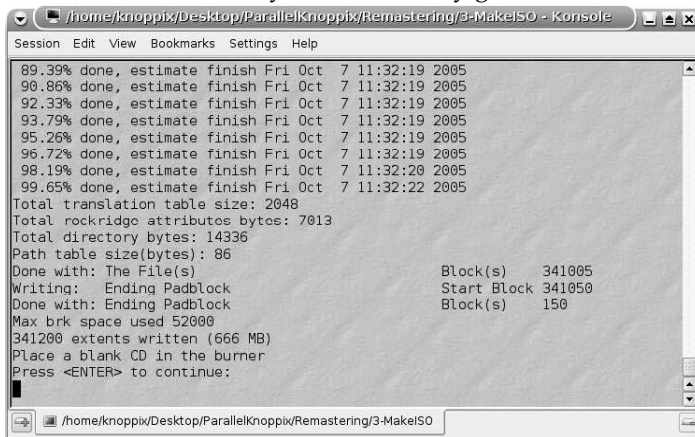
It asks you where you put the source for remaster:



Then it says that things appear to be in order, and asks if you want to proceed:



Quite some time later, you'll eventually get to this:



I recommend hitting <CTRL-C> here to abort, and burning the ISO image you have created later, once you have shut down P-KPX. The ISO image you have created will be in the `parallel_knoppix_remaster` directory of whatever partition you chose when you ran the `1-CopySourceForRemaster` script.

6 Security

Inside the cluster, the normal user can gain root access simply by typing `su`. From outside the cluster, one can login as root using `ssh` and the root password which was given above. The root user can gain read/write access to any partition on any computer in the cluster. A malicious user could view private data and could destroy it. It is clear that P-KPX is not a secure system. It is very easy to use even for people with no knowledge of clusters, but the ease of use is obtained at the expense of security. Administrators of systems should not let untrusted users boot the P-KPX CD, at least unless a good set disk images or some other backup system is in place. One should also be aware that connecting to the Internet while using P-KPX is not secure. Future work may be done to improve the security of P-KPX, but for now it is important to think of security issues before using it.

7 Conclusion

The ParallelKnoppix CD provides a very simple and rapid means of setting up a cluster of heterogeneous PCs of the IA-32 architecture for HPC parallel computing. It is not intended to provide a stable cluster for multiple users, rather is a tool for rapid creation of a temporary cluster for

individual use. The CD itself is personalizable, and the configuration and working files can be re-used over time, so it can provide a long term solution for an individual user.

I welcome comments, suggestions and contributions from anyone who uses this.

References

- [1] Creel (in press), User-friendly parallel computations with econometric examples, forthcoming in *Computational Economics*.
- [2] Creel (2005), A note on parallelizing the parameterized expectations algorithm, <http://econpapers.repec.org/paper/aubautbar/651.05.htm>.