ParallelKnoppix - Rapid Creation of a Linux Cluster for MPI Parallel Processing Using Non-Dedicated Computers

Michael Creel*

14th October 2004

Abstract

This note describes ParallelKnoppix, a bootable CD that allows creation of a Linux cluster in very little time. An experienced user can create a cluster ready to execute MPI programs in less than 10 minutes. The computers used may be heterogeneous machines, of the IA-32 architecture. When the cluster is shut down, all machines except one are in their original state, and the last can be returned to its original state by deleting a directory. The system thus provides a means of using non-dedicated computers to create a cluster. An example of maximum likelihood estimation done in parallel is provided.

Introduction

Linux clusters of relatively inexpensive workstation-class computers can achieve performance that is competitive with traditional specialized supercomputers. The Top500 (Top500 group, 2003) list of supercomputers, which is maintained by groups at the University of Mannheim, the University of Tennessee, and the National Energy Research Scientific Computing Center at the Lawrence Berkeley National Laboratory, is a periodically updated ranking of the power of supercomputers around the world. In the November 2003 listing, clusters of workstation-class computers running Linux, connected using specialized networking

*Department of Economics and Economic History, Edifici B Universitat Autònoma de Barcelona, 08193 Bellaterra (Barcelona) Spain. This research was supported by grants SGR2001-0164 and SEC2003-05112. I thank Javier Fernández Baldomero for his invaluable help in getting started with MPITB.

technology, occupied positions 4 through 7 in the ranking. Large arrays of similar but less expensive and lower performance computers (commodity computers) connected by ordinary ethernet cabling are commonly deployed at universities for the day-to-day work of students and employees. If these computers are organized as a Linux cluster, the available computing power will not be of the level of the top research clusters, but it will be much greater that that of an isolated desktop machine.

In fact, groups at many universities have created this sort of cluster. Such clusters often run at night when the computers are otherwise unoccupied. If the array of computers is homogeneous, setting up and maintaining a cluster is relatively straightforward, but it is a job that requires special training by information technology personnel. It is a job that must be repeated when the computers are replaced by new models, which may occur relatively frequently since universities often rent commodity computers. Configuration and security issues are not trivial. If many users are to access the cluster, installation and maintenance of the packages they need is another source of work for IT personnel. Some universities or departments may have a budget that allows achievement of a satisfactory level of support, but many groups may find that installation and maintenance of a serviceable cluster is infeasible. An array of heterogeneous computers with various network cards, CPU's, different hard disk partition layouts, *etc.*, is even more difficult to setup and maintain.

Likewise, researchers and other users of parallel computing may face difficulties in disseminating their work. Such work cannot easily be demonstrated if a cluster is not available, which may be the case at a conference or when visiting another institution. Even if a cluster is available, it may not have needed libraries installed, or they may be of incompatible versions.

There may be many potential users of parallel computing who do work that could be executed on a cluster. Many people will find that the detailed knowledge needed to create a cluster from scratch is overwhelming in the absence of support personnel to take care of the task, and they may never get beyond the stage of curiousity.

This note presents a solution that allows creation of a working Linux cluster on a network of computers of the IA-32 architecture (*e.g.*, Intel Pentium or Xeon, or AMD Athlon or

Duron computers) in a matter of minutes. The cluster can be made up of homogeneous or heterogeneous computers, and they need not have Linux installed. It is quite easy to use. It allows software packages and personal files to be added, so that individual users can tailor the solution to their needs.

Description

ParallelKnoppix (Creel, 2004) is a modification of the Knoppix (Knopper, undated) Linux distribution. It adds the LAM/MPI (LAM team, 2004) and MPICH (Gropp, *et. al.,* 1996) implementations of the MPI standard (Message Passing Interface Forum, 1997), which allow applications to run in parallel. The nodes of the cluster are configured almost automatically, and a working directory is NFS shared by all nodes. The CD is distributed pre-configured to support from 2 to 50 nodes, but it can easily be modified to allow for larger clusters.

ParallelKnoppix gives the user complete control over all of the nodes of the cluster. A user can easily delete or modify data on any hard disk partition of any of the nodes. As such, network administrators should not let untrusted users work with it, and it is advisable to have important data backed up in case of mistakes. ParallelKnoppix provides a very easy means of creating a cluster. The ease of setup is obtained largely at the expense of security. It is also advisable to isolate the cluster from external networks, since anyone that is knowledgeable about the details of ParallelKnoppix could easily gain access to any machine in the cluster.

How it works

Knoppix is a live Linux filesystem on a bootable CD. When booted, the hardware on the computer is automatically detected, and the system configures itself to run a modified version of Debian Linux.

ParallelKnoppix is a re-mastering of the Knoppix CD that has been configured to quickly create a Linux cluster with a modest amount of intervention. The master computer is booted using the CD, and hardware is detected automatically, just as with Knoppix. Then a ter-

minal server is launched, and the slave nodes are booted from their network cards. Each node then makes separate use of the automatic hardware configuration of Knoppix. Thus, the nodes can be heterogeneous. The basic requirement is that the nodes be booted from their network cards using PXE, or using a CD or floppy disk that may be obtained from http://rom-o-matic.net that simulates this ability. When all nodes are booted, a script is run. This script:

- creates a working directory on a storage device attached to the master node
- NFS exports the working directory from the master node
- mounts it on the nodes

At this point, the cluster is ready to run MPI programs using LAM/MPI or MPICH. The entire process takes less than ten minutes for an experienced user.

It is worth emphasizing again that ParallelKnoppix gives the user complete control over all of the nodes of the cluster. A user can easily delete or modify data on any hard disk partition of any of the nodes. As such, administrators should not let untrusted users work with it. It would also be advisable to have disk images or some other backup of all nodes available, in case a disastrous mistake is made. ParallelKnoppix provides a very easy means of creating a cluster. The ease of setup is obtained largely at the expense of security.

Use

A detailed tutorial with many screenshots that shows how to use ParallelKnoppix is available at

http://pareto.uab.es/wp/2004/62604.pdf

Personalization

Users will likely find that ParallelKnoppix does not contain packages that they need, and in any case they will need to access their own data and files.

Addressing the first issue, ParallelKnoppix is a remaster of Knoppix, and ParallelKnoppix itself may be remastered in the same way. Remastering is the process of copying the information on the CD to a hard disk, modifying the contents, and creating a new bootable CD image. Some discussion of how to do this and scripts that mostly automate the process are included on the CD, and there is extensive information on the topic available at

`http://www.knoppix.net/docs/index.php/KnoppixRemasteringHowto`

To add or remove software packages, the convenient `apt-get` tool that will be familiar to users of Debian Linux is available. Significant space (about 200MB) has been left free on the CD so that users have plenty of room to add packages and their own files without having to remove applications.

To add personal data and files, a number of means are available. If ParallelKnoppix is booted on a computer that contains the needed information, it may simply be read from the storage media and copied into the working directory. USB pen drives may also be used. It is also possible to access directories that have been NFS exported from other computers on the network.

Example: maximum likelihood estimation

To show that ParallelKnoppix can in fact create an efficient environment for MPI computing, a simple example will suffice. MPITB (Fernández Baldomero, 2004) for GNU Octave (Eaton, 1998) provides a collection of bindings to the LAM/MPI functions for MPI parallel processing that allows Octave programs to make use of MPI message passing to achieve parallelization.

We consider maximum likelihood estimation of a reshaped negative binomial density for count data. The negative binomial base density is multiplied by a squared polynomial, then normalized to sum to one. References that explain the approach are Gallant and Nychka (1987), Cameron and Johannson (1997) and Guo and Trivedi (2002). Since this is used only to provide an example of computational gains, we do not go into details here. The reshaped

density is

$$f_Y(y|\psi,\lambda,\gamma) = \frac{\left[h_p\left(y|\gamma\right)\right]^2}{\eta_p(\phi,\gamma)} \frac{\Gamma(y+\psi)}{\Gamma(y+1)\Gamma(\psi)} \left(\frac{\psi}{\psi+\lambda}\right)^{\psi} \left(\frac{\lambda}{\psi+\lambda}\right)^{y},$$

where

$$h_p\left(y|\gamma\right) = \sum_{k=0}^{p} \gamma_k y^k, \tag{1}$$

and

$$\eta_p(\psi,\lambda,\gamma) = \sum_{k=0}^{p}\sum_{l=0}^{p} \gamma_k\gamma_l m_{k+l}(\psi,\lambda) \tag{2}$$

In this last equation, the $m_{k+l}(\psi,\lambda)$ are the raw moments of the negative binomial density. Since this double sum leads to very long analytic expressions when p is at all large, evaluation of the density is relatively computationally intensive, and it may benefit from parallelization.

For n independent observations, the maximum likelihood estimator can be defined as

$$\hat{\theta} = \arg\max s(\theta)$$

where

$$s_n(\theta) = \frac{1}{n}\sum_{t=1}^{n} \ln f(y_t|x_t,\theta)$$

As Swann (2002) points out, this can be broken into sums over blocks of observations, for example two blocks:

$$s_n(\theta) = \frac{1}{n}\left\{\left(\sum_{t=1}^{n_1} \ln f(y_t|x_t,\theta)\right) + \left(\sum_{t=n_1+1}^{n} \ln f(y_t|x_t,\theta)\right)\right\}$$

Analogously, we can define up to n blocks. Again following Swann, parallelization can be done by calculating each block on separate computers.

The above model was estimated using 7930 observations on the number of doctor visits made by individuals, using Octave and MPITB to write a parallelized function to calculate the likelihood function. A cluster was created using ParallelKnoppix. The computers were homogeneous Pentium IV machines running at 2.8 GHz , each with 512MB of RAM and a 3COM 3c905 Tornado network card. They were connected on a 100MB/s ethernet network.

Table 1: Runtime as a function of number of nodes

Nodes	Runtime (seconds)
1	125.68
2	72.98
3	53.67
4	44.40
5	38.66
6	34.87
7	32.26
8	29.87
9	27.47
10	26.77

Table 1 shows the times needed to maximize the likelihood function using different numbers of computers, and Figure 1 plots them. In this fairly simple example, the runtime can be cut by a factor of almost 5 by using parallel computation. This example is not meant to be representative, and it is not even optimized. It is merely meant to show that ParallelKnoppix allows creation of a cluster that can achieve an important speedup in a real-world type problem.

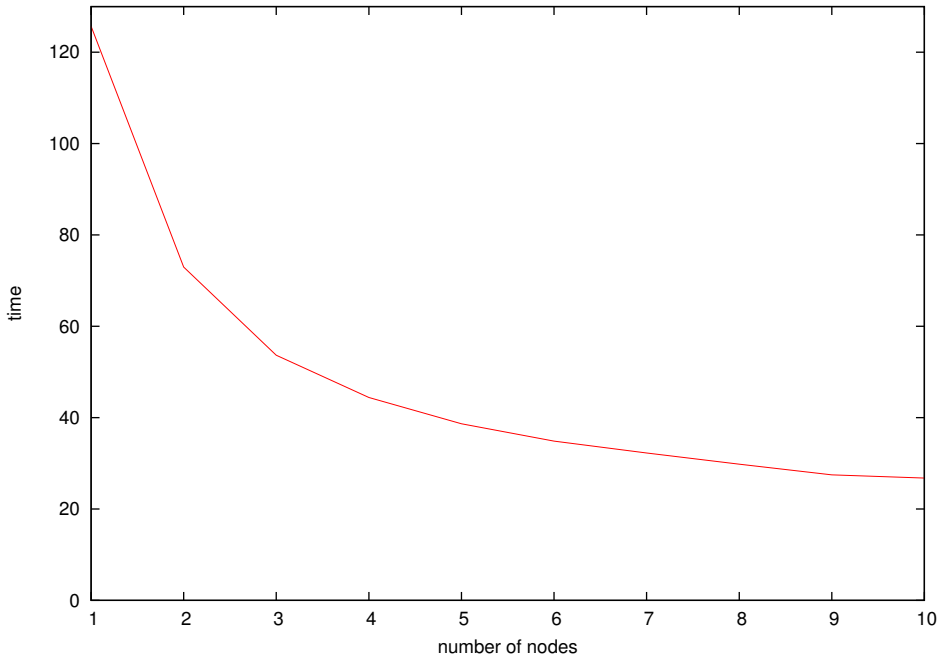
Conclusion

The ParallelKnoppix CD provides a very simple and rapid means of setting up a cluster of heterogeneous PCs of the IA-32 architecture. It is not intended to provide a stable cluster for multiple users, rather is is a tool for rapid creation of a 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. Suggestions and contributed improvements or examples are welcomed.

References

[1] Cameron, A.C. and P. Johansson (1997), Count data regression using series expansions: with applications, *Journal of Applied Econometrics,* **12**, 203-23.

Figure 1: Runtime as a function of number of nodes



[2] Creel, Michael (2004), "ParallelKnoppix - Create a Linux Cluster for MPI Parallel Processing in 15 Minutes", pareto.uab.es/mcreel/ParallelKnoppix/.

[3] Eaton, J.W. (1998), "Octave Home Page", www.octave.org

[4] Fernández Baldomero, J. (2004), "LAM/MPI Parallel Computing under GNU Octave", atc.ugr.es/javier-bin/mpitb.

[5] Gallant, A.R. and D.W. Nychka (1987), Seminonparametric maximum likelihood estimation, *Econometrica*, **55**, 363-90.

[6] Guo, J.Q. and P.K. Trivedi (2002), Flexible parametric models for long-tailed patent count distributions, *Oxford Bulletin of Economics and Statistics*, **64**, 63-82.

[7] Knopper, Klaus (undated), "KNOPPIX - Live Linux Filesystem on CD", www.knopper.net/knoppix/index-en.html.

[8] LAM team (2004), "LAM/MPI Parallel Computing", www.lam-mpi.org/.

[9] Message Passing Interface Forum (1997), ”MPI-2: Extensions to the Message-Passing Interface”, University of Tennessee, Knoxville, Tennessee.

[10] Gropp, W., E. Lusk, N. Doss and A. Skjellum (1996), "A high-performance, portable implementation of the MPI message passing interface standard", *Parallel Computing*, **22**, 789-828, see also www-unix.mcs.anl.gov/mpi/mpich/.

[11] Swann, C.A. (2002). ”Maximum likelihood estimation using parallel computing: an introduction to MPI”, *Computational Economics*, **19**, 145-178.

[12] Top500 group (2003), ”Top 500 Supercomputer Sites”, www.top500.org/list/2003/11/.