

Distributed Computing on Linux

Hertz Donors

A variety of projects with completely different goals are currently competing for the use of the latent processing power of home PCs. This article provides an overview of the more interesting efforts.

BY BJÖRN GANSLANDT

Even though the daily blurb from various computer and chip manufactures might suggest that your computer needs even more power, you will in fact very rarely need to tax your CPU to the limit. The average PC has only a moderate load most of the time. In addition to CPU cycles most PCs have some bandwidth to spare, allowing them to coordinate processing tasks with other PCs. In this way, millions of PCs can be linked up to work in parallel on jobs that would normally require a super computer at an exorbitant asking price.

Nearly all of the distributed computing projects discussed in this article are available for Linux in the form of *tar.gz* archives and can be unpacked in the usual way.

```
tar -zxvf archive.tar.gz
```

Since the client source code is typically not included, there is no need to compile it – just launch the client instead. But you could even leave that task to another program called *cron* – refer to the *Cron Setup* inset for more information.

Cylon Radio

Probably the most famous distributed computing project with over 3.9 million users is SETI@Home, [1]. SETI@Home has set itself the daunting task of searching for intelligent, extraterrestrial



lifeforms, no less. To this end, the SETI@Home client analyzes a 100 second recording with a bandwidth of 10 kHz from a radio-telescope in Puerto Rico seeking signs of intergalactic radio transmissions. As the telescope rotates in relation to possible extra-terrestrial radio sources, the client searches for a signal that matches a Gaussian beam pattern. Additionally, the software has to consider doppler effects, recognize pulsed signals, and come to terms with the increasing number of terrestrial transmissions. Due to the enormous amount of volunteers, SETI@Home can evaluate each package more than once, and thus eliminate

errors or attempted manipulation. And thanks to the sheer bulk of data evaluated, it is also possible to filter out radio signals that occupy constant positions in the sky – unfortunately most of these permanent signals turn out to be terrestrial. The SETI software normally runs in character-based mode on Linux, but you can stipulate the *-graphics* option to relay your results to the GUI version, *xsetiathome*. If you want to run SETI@Home permanently as a background task, you might like to try the *-nice 19* option, which reduces the client program's priority. Why "nice"? Well the program gets out of the way if other programs need more CPU cycles. Apart

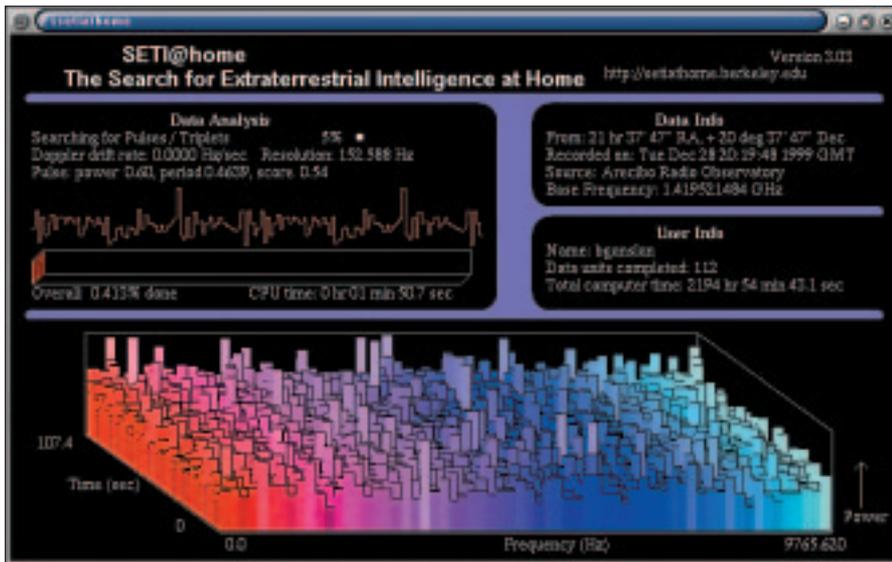


Figure 1: Xsetiathome visualizes the search for extraterrestrial radio signals

from *xsetiathome* there are a variety of other programs that convert the results to graphics, allowing you to insert them into the KDE panel. Try Freshmeat, if you are interested in finding a few [2].

Crack the Code

Distributed.net [3] are currently working in parallel on several mathematical tasks, and in contrast to SETI@Home they can point to a number of problems they have solved in the past. The project was able to crack DES and/or CSC encrypted messages in a record time. Currently Distributed.net are taking part in the RC5-64 competition – in contrast to other competitions, the participants are required to test a maximum of 2^{64} keys, compared to 2^{56} previously, and that certainly requires an enormous amount of processing power. You are more likely to be struck by lightning while winning the national lottery than find the right key with your first guess. However, you do get to keep US \$2,000 of the prize money, if you are the lucky finder. The rest of the prize money – which was sponsored by RSA – will go to Distributed.net, which is a non-profit organization, and if you are a member, to your local Distributed.net group.

The second active project at Distributed.net is the search for an Optimal Golomb Ruler with 24 or 25 integers, where the integers must be non-negative such that no two distinct pairs of numbers from the set have the same difference. An Optimal Golomb

Ruler is the shortest Golomb Ruler possible for a given number of marks. OGR's have many applications, including combinatorial functions, and in the field of interference phenomena.

Distributed.net also offers a console based client that does not impact your bandwidth or CPU cycles as heavily as SETI@Home. You can either configure the program after first launch or use the manual *-config* option. The configuration options allow you to set the priority for various projects and change the size of the work packages. You can also use the *-install* option to automatically add the program to */etc/init.d/* and assign the appropriate runlevel, allowing it to be launched whenever you start your computer – until you remove it, with the *-uninstall* option, that is.

More Maths: Prime Numbers

GIMPS [4] is another mathematical project, and they are looking for prime numbers this time – prime numbers of the form $2^p - 1$, where p is also a prime number, to be more precise. This type of prime number is referred to as a Mersenne prime number, named after the French monk and mathematician. The Electronic Frontier Foundation has put up a prize for the first prime number with at least 10 million digits, but GIMPS is not the only project seeking large prime numbers, and you need a lot of CPU cycles to find, or verify one.

The ECCp-109 project [5] has entered yet another cryptography competition

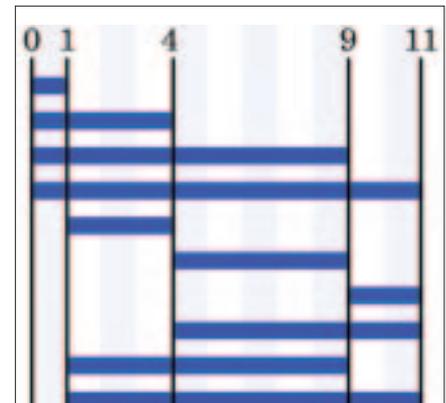


Figure 2: Any possible distances between two numbers on a Golomb Ruler must be of a different length

with slim chances of prize money. In contrast to RC5-64 this competition is not about symmetrical algorithms but an asymmetrical (Public Key) algorithm based on elliptic curves, where both a Public and a Private key exist. Encoding algorithms based on elliptic curves have the advantage of shorter keys and higher speeds when compared with traditional techniques like RSA or Elgamal, as used by PGP or GPG, however, more research is required on this subject.

Power Chess with Clusters

The success of computers such as Deep Blue or Deep Fritz (who has been battling it out with the reigning (BGN)

Cron Setup

Cron can be used to launch and terminate other programs at pre-defined times. You can use *crontab -e* to define tasks for the daemon. This command will load the editor defined in your *\$VISUAL* or *\$EDITOR* environment variable – use *export EDITOR=editor*, if you want to change this setting. The following entry launches a program at 8.00 pm every day and terminates the program at 9.00 am. The *2>&1 >/dev/null* string sends the program's output to the null device; cron would otherwise want to email this output to the user. If you use the "@reboot" parameter instead of specifying a schedule, cron will launch the designated program when you reboot your system – you can type *man 5 crontab* for additional information on the crontab format.

```
00 20 * * * cd mydirectory;
./CLIENT 2>&1 >/dev/null
00 9 * * * killall CLIENT
```

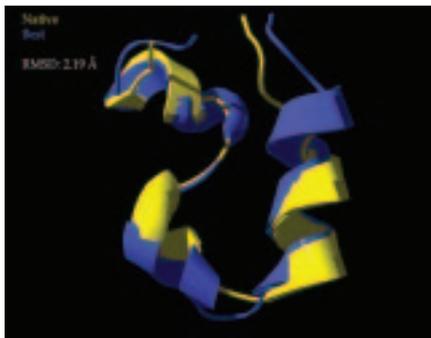


Figure 3: Predicting protein structures from distributedfolding.org

Chess world champion Wladimir Kramnik in October, assisted only by a team of eight professors) has shown that computers with a certain amount of processing power are extremely difficult to beat. The Chessbrain project, which is quite recent, is looking into the prospect of a powerful chess computer (see [6]) and has recently reached the first of four designated development stages. Chessbrain will not become a really powerful competitor until it reaches phase 3 – the work currently in progress primarily concerns the distributed infrastructure. One of the most fascinating aspects of this project is the use of the SOAP protocol to transfer data to the clients, or so-called PeerNodes. As SOAP can now be processed by FlashMX, Chessbrain not only offers the PeerNode software, but also various viewers based on Flash or PHP, for example, that allow you to view the current game. However, you will need a flat rate if you intend to sign up for this project, as the PeerNode continually accesses the server.

Proteins

Super computers are also important to medicine and play a vital role in the field

INFO

- [1] <http://setiathome.ssl.berkeley.edu>
- [2] <http://freshmeat.net>
- [3] <http://distributed.net>
- [4] <http://www.mersenne.org/prime.htm>
- [5] <http://www.nd.edu/~cmonico/eccp109/>
- [6] <http://chessbrain.net>
- [7] <http://folding.stanford.edu>
- [8] <http://gah.stanford.edu>
- [9] <http://www.distributedfolding.org>
- [10] <http://www.electricsheep.org>

of genetic research, such as the quest to decrypt the human genome. Research into proteins and corresponding genetic sequences is the aim of two partner projects: Folding@Home [7] and Genome@Home [8], which recently united to form a single client. Folding@Home is specifically concerned with the folding process of proteins, whose encryption could mean a breakthrough both in medicine and in nanotechnology. Genome@Home works with known protein structures and attempts to calculate appropriate, synthetic genetic sequences that allow genetic researchers to gain a better understanding of natural genetic sequences. Although Genome@Home has been integrated in the Folding@Home-Client, you can specify a Genome@Home team number (over 100.000) to work exclusively on the former project. Additionally, the original client is still available as Genome@Home Classic. However, before you can run the integrated client, you will need to make it executable by typing:

```
chmod +x FAH3Console-  
v312-Linux.exe
```

Another project working on proteins is Distributed Folding [9]. The procedure here is different to that followed by Folding@Home – the focus of Distributed Folding is on predicting protein structures, rather than folding. The folding process is particularly relevant to diseases such as Alzheimer or Creutzfeldt Jacob, that may occur in the context of proteins whose folding characteristics deviate from the norm.

Although most of the projects discussed so far are available as (more or less attractive) Windows screensavers, Linux users normally have to be content with boring text based interfaces that only occasionally issue a cryptic comment on the progress they are making. The Electric Sheep screensaver [10], which was inspired by Philip K. Dick's book "Do

Androids Dream of Electric Sheep" – as was the movie "Blade Runner" – is a notable exception. Of course computers don't count normal sheep, but instead use their processing power to create animated, fractal flames. In contrast to the other projects discussed, Electric Sheep provides both the source code and RPMs, allowing you to install the screensaver directly to your GNOME control center. If you do not use GNOME, you can type the following line to add Electric Sheep to the "programs:" section of "~/.xscreensaver":

```
"ElectricSheep"  
electricsheep \n\
```

The animations are sent to a central server that then returns the animation as an MPEG video to Electric Sheep screensavers all over the globe. Unfortunately, the volume of traffic involved restricts usage to those fortunate enough to have a DSL flat rate or similar Internet link. ■



Figure 4: Electric Sheep calculates fractal flames