Surviving on Planet CCRMA, two years later and still alive

Fernando Lopez-Lezcano
CCRMA, Stanford University
The Knoll, Stanford, CA 94305, USA
nando@ccrma.stanford.edu
http://ccrma.stanford.edu/planetccrma/software/

Abstract

Planet CCRMA at Home is a collection of packages that you can add to a computer running RedHat 9 or Fedora Core 1, 2 or 3 to transform it into an audio workstation with a low-latency kernel, current ALSA audio drivers and a nice set of music, midi, audio and video applications. This presentation will outline the changes that have happened in the Planet over the past two years (since the previous presentation at LAC2003).

Keywords

Audio Distribution, Package Collection, Low latency.

1 Introduction

Creating worlds is not an easy task, and Planet CCRMA is no exception. The last two years have seen a phenomenal expansion of the project. The history of it will reflect, I hope, part of the recent history of Linux Audio projects and kernel patching.

2 A bit of history

For those of you that are not familiar with Planet CCRMA a bit of history is in order. At CCRMA (the Center for Computer Research in Music and Acoustics at Stanford University) we have been using Linux as a platform for research and music production since the end of 1996 or so. Besides the software available in the plain distribution I installed at the time, I started building and installing custom music software in our main server (disk space was not what it is today, and there not that many Linux machines – at that time we were dual booting some PCs between Linux and NEXTSTEP, which was the main computing platform at CCRMA). I don't need to say that sound support for Linux in 1997 was a bit primitive. Not many sound cards were suppored, and very few existed that had decent sound quality at all. Low latency was not a concern as just getting reliable sound output at all times was a bit of a challenge. Eventually the sound drivers evolved (we went through many transitions, OSS, ALSA 0.5 and then 0.9), and patches became available for the Linux kernel that enabled it to start working at low latencies suitable for realtime reliable audio work, so I started building custom monolithic kernels that incorporated those patches and all the drivers I needed for the hardware included in our machines (building monolithic kernels was much easier than trying to learn the details of loadable kernel modules :-).

But over time hard disks became bigger so that there was now more free space in the local disks, and the number of Linux machines kept growing, so the server installed software was going to become a network bottleneck.

Also, some adventurous CCRMA users started to install and try Linux in their home machines, and wanted an easy way to install the custom software available in all CCRMA workstations.

I was installing RedHat so I started to use RPM (the RedHat Package Manager) to package a few key applications that were used in teaching and research (for example the Snd sound editor, the CM – CLM - CMN Common Lisp based composition and synthesis environment, Pd).

At first I just stored those packages in a network accessible directory and told people, “there you are, copy the rpm packages from that directory and install them in your machine”. A simple Web site was the next step, and installation instructions were added as I got feedback from users on problems they faced when trying to install the packages. Finally the project was made “public” with a post announcing it in the Cmdist mailing list – an email list for users of Snd and CM/CLM/CMN (although I later learned that some users had discovered the existence of the packages through search engines, and were already using them). The announcement happened on September 14th 2001. Time flies.

This changed the nature of the project. As more people outside of CCRMA started using the packages I started to get requests for packaging music software that I would not have thought of installing at CCRMA. The number of packages started to grow and this growth benefited both CCRMAlites and external Planet CCRMA users alike.

As the project (and this was never an “official” project, it was a side effect of me packaging stuff to install at CCRMA) grew bigger the need for a higher level package management solution became self-evident. The dreaded “dependency hell” of any package based distribution. More and more packages had external dependencies that had to be satisfied before installing them and that needed to be automatic for Planet CCRMA to be really usable. At the beginning of 2002 apt for rpm (a port of the Debian apt tool by Conectiva) was incorporated into Planet CCRMA, and used for all package installation and management. For the first time Planet CCRMA was reasonably easy to install by mere mortals (oh well, mere geek mortals).

Fast forward to today: there are more than 600 individual packages spanning many open source projects in each of the supported branches of RedHat/Fedora Core. You can follow the external manifestation of these changes by reading the online ChangeLog that I have maintained as part of the project (a boring read, to say the least).

3 At the core of the Planet

Since the official announcement of the project outside CCRMA on September 2001, the base distribution on which it was based (RedHat) has seen significant changes. In July 2003 RedHat stopped releasing commercial consumer products, and the last RedHat consumer version was 9, released on March 2003. The Fedora Project was created, with the aim of being a community driven distribution with a fast release cycle that would also serve as a testbed for new technologies for the enterprise line of RedHat products. Fedora Core 1 was the first release, followed by Fedora Core 2 and 3, at approximately 6 month intervals. The rapid release cycle plus the introduction of new technologies in the releases have made my life more “interesting”.

In particular, Fedora Core 2 saw the introduction of the 2.6 kernel, which created a big problem for rebuilding the Planet CCRMA package collection on top of it. The core problem: a good, reliable low latency kernel did not exist. At that point in time 2.6 did not have an adequate low latency performance, despite the assurances heard during the 2.5 development cycle that the new infrastructure in the kernel was going to make it possible to use a stock kernel for low latency tasks. Alas, that was not possible when Fedora Core 2 was released (May 2004).

3.1 The kernels

Up to Fedora Core 1 the base distribution used a 2.4 kernel, and Planet CCRMA provided custom kernel packages patched with the well known low latency (by A. Morton) [6] and preemptible kernel (by R. Love) [5] patches (the last originally by Monta Vista [4]), in addition to the tiny capabilities patch that enabled to run the Jack

Audio Connection Kit and his friends with realtime privileges as non-root users.

Fedora Core 2 changed the equation with the introduction of the 2.6 kernel. Running a 2.4 kernel on top of the basic distribution presented enough (small) compatibility problems that I discarded the idea very early in my testing cycle. And 2.6 had a very poor latency behavior, at least in my tests. As a consequence until quite recently I still recommended using Fedora Core 1 for new Planet CCRMA installs.

For the first 2.6 kernels I tested (March 2004) I used a few additional patches by Takashi Iwai [7] that solved some of the worst latency problems. But the results were not very usable.

Ingo Molnar and Andrew Morton attacked the problem and a very good solution evolved that is now available and widely used. Ingo started writing a series of patches for realtime preemption of the 2.6 kernel [8] (named at the beginning the “voluntary preemption” patchset). This set of patches evolved on top of the “mm” patches by Andrew Morton [9], the current equivalent of the old unstable kernel series (there is no 2.7 yet!, experimental kernel features first appear in the “mm” patches and then slowly migrate – the successful ones, that is - to the official release candidates and finally to the stable releases of the Linux kernel). Ingo did very aggressive things in his patches and the voluntary preemption patches (later renamed realtime preemption patches) were not the most stable thing to run in your computer, if it booted at all (while tracking successive releases I must have compiled and tried out more than 40 fully packaged kernels, for details just look at the changelog in the spec files of the Planet CCRMA 2.6 kernels). I finally released a preliminary set of kernel packages on December 24 2004, using version 0.7.33-04 of Ingo's patches, one of the first releases that managed to boot in all my test machines :-)

What proved out to be interesting and effective in Ingo's patches gradually percolated to the not so bleeding edge “mm” patches by Andrew Morton, and bits and pieces of “mm” gradually made it upstream to the release candidates and then to the stable kernel tree.

So, little by little the latency performace of the stock kernel improved. By the time of the release of 2.6.10 (December 24 2004 again – just a coincidence) it was pretty good, although perhaps not as good as a fully patched 2.4 kernel. But keep in mind that this is the stock kernel with no additional patches, so the situation in that respect is much much better than in the old stock 2.4 kernel.

The end result for Planet CCRMA dwellers at the time of this writing are two sets of kernels, currently available on both FC2 and FC3.

3.1.1 The “stable” kernel

The current version is 2.6.10-2.1.ll. 2.6.10 turned out to be an unexpected (at least by me) milestone in terms of good low latency behavior. Finally, a stock kernel that has good low latency performance, out of the box. I would say it is close to what a fully patched 2.4 kernel could do before. The package also adds the realtime lsm kernel module, more on that later.

3.1.2 The “edge” kernel

Currently 2.6.10-0.6.rdt based on version 0.7.39-02 of Ingo's patch. This is a more bleeding edge kernel, with significantly better low latency performance and based on Ingo Molnar's realtime preemption patches. The downside of trying to run this kernel is that it still (at the time of this writing) does not work perfectly in all hardware configurations. But when it works, it works very well, and users have reported good performance with no xruns running with two buffers of 64 or even 32 samples! Amazing performance.

I'm still being a bit conservative in how I configure and build this kernel as I'm not currently using the REALTIME_RT configuration option, but rather the REALTIME_DESKTOP option (thus the rdt in the release). The penalty in low latency behavior is worth the extra stability (at this time). I hope that the RT option (which gets the linux kernel close to being a “hard realtime” system) will evolve and become as stable as the REALTIME_DESKTOP configuration.

These packages also include the realtime lsm module.

3.1.3 Small details that matter

But a kernel with good low latency is not nearly enough. You have to be able to run, for example, Jack, from a normal non-root account. Enter Jack O'Quinn [9] and Torben Hohn. Their efforts created a kernel module, part of the kernel security infrastructure, that enables applications run sgid to a certain group, or run by users belonging to a group, or run by any user (all of this configurable, even at runtime), to have access to realtime privileges without having to be root. This is more restrictive and secure than the old capabilities patch, and at the time of this writing and after a very long discussion in the Linux Kernel mailing list (see [11] and [12]), has been incorporated into the “mm” kernel patches. Hopefully it will eventually percolate down to the stable kernel tree at some point in the future. It was a tough sell advocating for it in the Linux Kernel mailing list, many thanks to Jack O'Quinn and to Ingo Molnar and Con Kolivas for proposing workable alternatives (that were later discarded). When the realtime patch becomes part of the standard kernel tree, a stock kernel will not only have decent low latency performance but will also work with software that needs realtime privileges like Jack does (including the ability of applications to run with elevated SCHED_FIFO scheduling privileges and to lock down memory so that it is not paged to disk).

But this was not enough for a Planet CCRMA release. Ingo Molnar's realtime preemption patch changed the behavior of interrupt requests, the lower half of the interrupt processes (if I understand correctly) are now individual processes with their own scheduling class and priorities, and a vital part of the tuning a system for good low latency behavior is to give them, and Jack itself, the proper realtime priorities so that the soundcard and its associated processes have more priority than other processes and peripherals. I was trying to find a solution to this that did not involve users looking around /proc and tuning things by hand, when Rui Nuno Capela sent me a neat startup service script called rtirq that does just that, it sorts all interrupt s ervice routines and assigns them decent priorities. Together with another small startup script I wrote that loads and configures the realtime lsm module, they make it possible to package an easy to install turn-key solution to a low latency 2.6 based kernel.

3.1.4 The core packages

The end result in Planet CCRMA are two sets of meta packages that reduce the installation and configuration of a 2.6 kernel to two apt-get invocations (installing planetccrma-core for the safer kernel and planetccrma-core-edge for the more risky one that offers better low latency performance).

This, coupled to the fact that due to 2.6 both Fedora Core 2 and 3 use ALSA by default, made installing Planet CCRMA is a much easier process when compared to Fedora Core 1 or RedHat 9 and 2.4 kernels.

4 Continents and islands

A large number of applications and supporting libraries have been added and updated over time to Planet CCRMA since LAC2003. Although not as many as I would like (just take a look at the “Pipeline” web page for packages waiting to be added to the repository). The list is almost too long but here it goes (starting from where I got back from LAC to CCRMA on 2003): seq24, filmgimp (later renamed to cinepaint), fluidsynth (formerly iwusynth), the mcp ladspa plugins, hydrogen, rezound, cinelerra, mammut, csound, qarecord, qamix, qjackctl, gmorgan, ceres, pmidi, denemo, jackeq, cheesetracker, rev ladspa plugins, qsynth, xmms-jack, jamin, vco ladspa plugins, pd externals (percolate, creb, cxc, chaos, flext, syncgrain, idelay, fluid, fftease, dyn), tap ladspa plugins, timemachine, caps ladspa plugins, xmms-ladspa, specimen, simsam, pvoc, brutefir, aeolus, fil ladspa plugins, pd vasp externals, jaaa, tap reverb editor, jackmix, coriander, liblo, jack bitscope, dvtitler, soundtouch library, beast, phat, sooperlooper, qmidiarp, dssi. Sigh, and that's only new packages. Many many significant updates as well.

5 Other worlds

Planet CCRMA is one of many package repositories for the RedHat / Fedora family of distributions. Freshrpms, Dag, Atrpms, Dries and many others constitute a galaxy of web sites that provide easy to install software. Planet CCRMA is in the process of integrating with several of them (the so called RpmForge project) with the goal of being able to share spec files (the building blocks of RPM packages) between repositories. That will make my work, and that of the other packagers, easier, will reduce the inevitable redundancy of separate projects and will increase compatibility between repositories.

Another world in which I also want to integrate parts of Planet CCRMA is the Fedora Extras repository. This Fedora sponsored project opened its first CVS server a short while ago and will be a centralized and more “official” repository of packages, but probably exclusively dedicated to augmenting the latest Fedora Core release. The “community” part of the Fedora Project is finally arriving and I'm looking forward to becoming a part of it.

6 A Planet Forge

A short time ago I finally got all the remaining components, and finished building a new server here at CCRMA. It is a fast dual processor machine with a lot of memory and hard disk space completely dedicated to the Planet CCRMA project. The original goal was to create a fast build machine in which to queue packages to be rebuilt, as that process was fast becoming one of my main productivity bottlenecks in maintaining Planet CCRMA. A secondary, but no less important goal, is to try to create a collaborative environment in which more people could participate in the development and maintenance of Planet CCRMA packages. We'll see what the future brings. A lot of work remains to be done to port my current build environment to the new machine and create a collaborative and more open environment.

7 Future directions

One of the many things that are requested from time to time in the Planet CCRMA lists is the mythical “single media install” of Planet CCRMA (ie: “do I have to download all these cdroms?”). In its current form (and on purpose), a potential user of Planet CCRMA has to first install Fedora Core, and then add the kernel, drivers and packages that make up Planet CCRMA (this additional installation and configuration work has been substantially reduced in Fedora Core 2 and 3 releases as they use ALSA instead of OSS). While this is not that hard, specially with the help of meta packages and apt-get (or synaptic), it appears that sometimes it is too much work :-) And I have to agree, it would be much nicer to have a single cd (hmm, actually a dvd given the size of current distributions) and at the end of the install have everything ready to go, low latency kernel active, just start the applications and make some music. I have long avoided going down this road and becoming a “distribution” because of the additional work that would involve. It is hard enough trying to keep up to date with the very fast evolution of Linux audio software.

But on and off I've been thinking about this idea, and lately I've been actually doing some stuff. At the time of this writing (end of February 2005) I already have a single “proof of concept” dvd with everything in it, all of Fedora Core 2 - the distro I've been playing with, I obviously have to do this on Fedora Core 3 as well - plus all of Planet CCRMA. This test dvd is not small, about 3G of stuff, remember, all of Fedora Core is included!

Installing Planet CCRMA entails booting into the dvd, selecting the Planet CCRMA installation target, customizing the packages installed if desired and pressing “Install” (while going through the normal installation choices of a stock Fedora Core system install, of course). One reboot and you are up and running. Furthermore, the dvd creation process is pretty much automatic at this time (start a series of scripts, wait for some time and out comes a dvd iso image).

Of course things are not that easy. What kernel should I select for installation? The more stable or the more risky that has better latency performance? How will the idiosyncracies of this non-standard kernel interact with the Fedora Core install process? (for example, it may happen that it will fail to boot in some machines, while the original Fedora Core kernel would have succeeded – and I don't think Anaconda, the RedHat installer, would be able to deal with more than one kernel at install time). Hopefully some or all of these questions will have answers by the time I attend LAC2005, and conference attendees will be able to test drive an “official” alpha release of Planet CCRMA, the distro (another question to be answered: why do I keep getting into a deeper pit of support and maintenance stuff??).

8 Conclusion

It is easy to conclude that Planet CCRMA is very cool. More seriously. Planet CCRMA as a project is alive and well. As a maintainer I'm (barely) alive, but have made it to another LAC conference, no small feat.

9 Acknowledgements

The Planet CCRMA project would never have been possible without the support for GNU/Linux and Open Source at CCRMA, Stanford University, and in particular, Chris Chafe, CCRMA's Director. It goes without saying that I extend my heartfelt thanks to the hundreds of commited developers whose software projects I package. Without them Planet CCRMA would not exist and I would live in much more boring world.

References

[1] The Fedora Project:
http://fedora.redhat.com/

[2] Planet CCRMA:
http://ccrma.stanford.edu/planetccrma/software/

[3] Ingo Molnar: Low latency patches for 2.2/2.4
http://people.redhat.com/mingo/lowlatency-patches/

[4] MontaVista: The Preemptible Kernel Path
http://www.mvista.com/
http://www.linuxdevices.com/news/NS7572420206.html

[5] Robert Love: The Preemptible Kernel Patch
http://rlove.org/linux/

[6] Andrew Morton: Low latency patches for 2.4
http://www.zip.com.au/~akpm/linux/schedlat.html

[7] Takashi Iwai: low latency tweaks
http://kerneltrap.org/node/view/2702

[8] Ingo Molnar: Realtime Preemption patches for 2.6
http://people.redhat.com/mingo/realtime-preempt/

[9] Andrew Morton: the “mm” patches for 2.6
http://kernel.org/pub/linux/kernel/people/akpm/patches/2.6/

[10] Jack O'Quinn: the realtime lsm kernel module
http://sourceforge.net/projects/realtime-lsm/

[11] Linux Weekly News: Merging the realtime security module
http://lwn.net/Articles/118785/

[12] Weekly News: Low latency for Audio Applications
http://lwn.net/Articles/120797/