

Audiophilie linuxienne

Pour de nombreux mélomanes et audiophiles, l'engouement populaire pour les mp3 et les baladeurs associés a de quoi faire sourire. Un mp3 à 192kb/s est souvent perçu comme de piètre qualité, surtout quand on compare son rendu sonore à celui d'un CD lu sur matériel dit audiophile. La compression avec perte des formats comme mp3 ou ogg est telle qu'elle finit par devenir gênante. Bien entendu, sur un baladeur numérique ou sur les enceintes bas de gamme d'un PC, la différence est beaucoup plus difficile à percevoir, mais de nombreux audiophiles n'écoutent de musique que sur leur matériel préféré.

Les formats numériques compressés avec perte possèdent cependant de nombreux avantages. Outre la réduction importante de place (le débit d'un CD sans compression est d'environ 1378kb/s), ces formats de fichier sont enrichis par des métadonnées (les *tags*) qui décrivent le contenu de l'extrait musical (compositeur, interprètes, titre, numéro de piste, etc.). Une bibliothèque de fichiers musicaux numériques s'organise facilement grâce à des outils comme **amarok**, ce qui enrichit l'expérience musicale (voir l'article dédié dans ce numéro). Moyennant un investissement raisonnable en capacité de stockage, on peut stocker des dizaines de CD sur son ordinateur et profiter ainsi d'une facilité d'accès inégalée.

Tous ces avantages de la représentation numérique de la musique sont très tentants pour le mélomane, mais la piètre qualité musicale du matériel PC reste une barrière. Pourtant, en combinant des logiciels libres et un dispositif matériel spécifique, on peut obtenir une qualité musicale exceptionnelle tout en stockant sa musique de façon numérique. La solution retenue dans cet article s'organise de la façon suivante :

- * la musique est extraite du CD de la façon la plus propre possible grâce au logiciel libre **cdparanoia** ;
- * la musique numérique est stockée sur un serveur **linux** au format **flac**, un format libre et ouvert de compression sans perte ;
- * la musique numérique est organisée sur le serveur grâce à **slimserver**, un logiciel libre qui peut entre autres envoyer un fichier musical sous forme de flux réseau ;
- * la musique numérique est convertie en signal analogique par une **squeezebox** (produit de la société **slimdevices**), un dispositif matériel qui s'interface avec **slimserver** pour récupérer la musique sur le serveur et la transmettre à un amplificateur (ou d'ailleurs à un convertisseur numérique/analogique (DAC)externe).

Chaque élément de cette chaîne joue un rôle important dans la qualité musicale obtenue au final. Grâce à **cdparanoia**, on évite les erreurs de lecture sur le CD et on récupère ainsi un fichier musical (au format **wave**) identique (ou presque) à celui qui a été enregistré lors de la prise de son. Flac assure une compression sans perte, ce qui limite la place utilisée tout en conservant une qualité sonore strictement identique à celle de l'original. De plus, le format flac stocke les métadonnées nécessaires à une bonne organisation de la bibliothèque musicale. Enfin, la conversion numérique analogique réalisée par la **squeezebox** donne un rendu sonore impossible à obtenir avec la carte son d'un PC (par exemple). Depuis sa version 2, la **squeezebox** est en effet reconnue dans le monde audiophile comme une solution d'excellente qualité. Le rapport qualité/prix est particulièrement exceptionnel dans le monde audiophile car la **squeezebox** coûte moins de 350 €. Un lecteur de CD audiophile coûte au moins 3 fois plus (il semble qu'il n'existe pas de limite haute) sans offrir les mêmes services.

Stockage numérique

La première étape de notre solution consiste donc à stocker sur un serveur **linux** notre musique au format **flac**. Comme **flac** compresse sans perte, il ne faut pas s'attendre à une économie de place très importante, mais plutôt prévoir 300 à 350 Mo par CD. Grossièrement, **flac** divise la place occupée par un morceau par deux, par rapport à l'occupation d'un fichier **wave**. Le débit est de l'ordre de 700kb/s. A titre d'exemple, une bibliothèque d'environ 600 CD occupe 160 Go. Un disque dur de 200 Go (et d'une capacité réelle de 200 milliards d'octets soit environ 186 Go) coûte moins de 100 € et permet ainsi de stocker approximativement 600 heures de musique au format **flac**. Par rapport au coût d'un CD, le stockage est donc relativement bon marché : si on considère un coût de 15 € pour un CD d'une heure, le stockage revient 90 fois moins cher que le CD lui-même. Bien entendu, on est loin des capacités induites par une compression **ogg** à 192kb/s qui permettrait le stockage de plus de 2 000 heures de musique sur le même disque dur.

Extraction (rip)

Pour transformer le CD en fichier flac, on dispose sous **linux** d'un outil très pratique, le logiciel **grip** (voir l'article dédié dans ce numéro). Celui-ci enchaîne automatiquement l'acquisition des pistes du CD au format **wave** avec **cdparanoia** puis la conversion en **flac**. Grâce à **cdparanoia** on obtient un fichier **wave** aussi parfait que possible en contournant les limites des lecteurs (et les bugs de leurs firmwares). Son principal outil est une lecture répétée des secteurs, ce qui le rend relativement lent par rapport aux vitesses de lectures annoncées par les constructeurs. Le résultat est cependant comparable à celui obtenu, grâce à des techniques similaires, par les lecteurs de CD audiophile (et hors de prix).

Conversion en flac

Le format wave utilisé par **cdparanoia** a deux défauts pour notre application : il n'est pas compressé et occupe donc grossièrement 1378kb par seconde de musique ; il ne contient aucune métadonnée. Le premier défaut n'est pas dramatique car les disques durs ont un coût au giga octets très faible par rapport au coût d'un CD. Par contre, le second défaut est très gênant car il impose l'emploi de solutions lourdes passant par le système de fichiers (nom et arborescence) pour identifier le contenu d'un morceau. En outre, il n'existe aucun standard de stockage et les applications de lecture sont donc incapables d'afficher des informations élémentaires comme le compositeur d'une oeuvre.

La solution audiophile passe par le format **flac** (*Free Lossless Audio Codec*). Ce format ouvert est l'oeuvre de Josh Coalson. Il s'agit d'abord de la spécification totalement ouverte d'un format numérique pour stocker de la musique sans perte, dont les caractéristiques sont les suivantes :

- * une grande souplesse : le format supporte par exemple de 1 à 8 voix par flux (un CD comporte 2 voix), de 4 à 32 bits par échantillon (un échantillon de CD comporte 16 bits) et une fréquence d'échantillonnage de 1 Hz à 655 350 Hz (celle d'un CD est de 44 100 Hz).

- * une adaptation à la transmission par le réseau : le format est organisé en *frames* indépendantes, le décodage d'une *frame* est possible sans connaître les autres. De plus chaque *frame* contient un CRC pour détecter d'éventuels problèmes de transmission. Enfin, le format est indexé ce qui permet de chercher rapidement une *frame* particulière ;

- * un contrôle d'intégrité : outre les CRCs, le format contient la signature **MD5** de l'original pour simplifier la vérification de l'intégrité du contenu sonore ;

- * des métadonnées très souples : les métadonnées sont stockées dans des blocs spécifiques et typés. Il est possible de définir de nouveaux types de métadonnées sans engendrer de problème pour les applications qui ne les utilisent pas. Les types standards incluent en particulier les *tags* qui décrivent la musique (compositeur, titre, etc.).

Le format spécifie aussi la méthode de compression, qui, outre son caractère sans perte (la décompression du fichier donne un contenu totalement identique à l'original), est adaptée à une implémentation matérielle.

Le système est en effet dissymétrique : la compression est beaucoup plus coûteuse en ressources que la décompression. Comme l'implémentation de référence de **flac** est fournie sous une licence très souple (la licence **Xiph**), son utilisation commerciale ne pose aucun problème, même dans des *firmwares* dont les sources ne sont pas distribuées (les algorithmes employés par flac ne sont protégés par aucun brevet connu). De ce fait, flac est supporté par de nombreux matériels comme la **squeezebox**, mais aussi des systèmes audiophiles comme les serveurs musicaux AudioReQuest et iMuse par exemple. Plus étonnant, alors que le format ne comprend aucun DRM (Digital Restrictions/Rights Management), des artistes comme Metallica et des labels comme Magnatune n'hésitent pas à diffuser leurs oeuvres à ce format.

Grâce à **grip**, la conversion des fichiers **wave** en fichier **flac** se fait de façon transparente. Il suffit de configurer l'onglet codage de grip de la façon suivante :

* dans la zone « Programme de codage », il faut préciser le chemin d'accès complet à **flac**, généralement **/usr/bin/flac** ;

* dans la zone « Ligne de commande du codeur », il faut passer à flac les paramètres lui permettant d'intégrer les métadonnées récupérées sur freedb, comme par exemple

```
-V --best -T title="%n" -T artist="%a" -T album="%d" -T date="%y" -T tracknumber="%t" -T genre="%G" -o %m %w
```

Décodons ces options :

* **-V** demande à **flac** de vérifier pendant l'encodage que le fichier se décode bien de façon strictement identique à l'original (utile pour les paranoïaques) ;

* **--best** correspond à la compression la meilleure possible (et donc à la plus lente à réaliser, sans impact sur la vitesse de décodage) ;

* les différentes options **-T** précisent les métadonnées à intégrer au fichier ;

* **-o** précise le fichier de sortie et le dernier paramètre est le fichier d'entrée (au format **wave**).

* dans la zone « Format du fichier codé », on indique le nom complet du fichier flac, en utilisant, comme dans les options de flac, les variables de **grip** (%n pour le titre de la piste, par exemple).

Serveur de fichiers musicaux

Une fois les fichiers musicaux stockés sur notre machine **linux**, il nous faut une solution pour les exploiter. Sur le serveur ou sur d'autres machines sur l'espace de stockage est exporté (par **nfs** ou **samba**, par exemple), de nombreuses solutions sont envisageables, comme **xmms** ou **amarok**. Cependant, cela nécessite d'exploiter la carte son du PC, soit comme DAC, soit comme transport, c'est-à-dire en exploitant la sortie numérique de la carte (coaxiale ou optique) afin de nourrir un DAC externe. Cela implique entre autres la présence du PC à proximité de l'amplificateur et des enceintes. Pour ne pas perturber l'écoute, il faut donc un PC silencieux. Pour un audiophile, la seule idée d'utiliser la carte son du PC comme source a de quoi faire trembler : l'intérieur d'un PC est une grosse soupe de bruit électronique et la qualité sonore de la conversion numérique/analogique d'une carte son est assez faible. Les sorties numériques sont aussi d'assez mauvaises qualités (essentiellement pour des raisons d'instabilité des horloges utilisées).

La solution pour un audiophile est donc d'utiliser un matériel dédié, capable de se connecter au serveur pour recevoir les fichiers et les convertir proprement en un signal sonore. Notre client est ici la **squeezebox** qui nécessite un logiciel serveur spécifique à faire tourner sur le serveur : **slimserver**.

Le logiciel **slimserver**, programmé en **Perl**, est libre (sous licence **GPL**) et par nature multi-plateforme. Sous linux, il est disponible sous forme d'un **rpm** ou des sources.

L'installation ne pose généralement aucun problème, sauf sous les dernières versions de **Mandriva**. Sous cette distribution, certains modules Perl semblent incompatibles avec **slimserver**. Heureusement, le logiciel peut installer des versions « standard » en les téléchargeant sur le **CPAN**. Pour ce faire, il suffit de lancer le programme

```
/usr/local/slimserver/Bin/build-perl-modules.pl
```

qui télécharge et compile les modules nécessaires. **Slimserver** fonctionne ensuite parfaitement.

Slimserver se pilote et se configure par une interface web très simple. Sa principale fonction est d'envoyer les fichiers audio depuis le serveur de stockage vers la **squeezebox**. Il dispose des capacités suivantes :

- * indexation de l'ensemble des fichiers musicaux grâce aux métadonnées contenues dans les fichiers : on peut alors chercher un morceau et naviguer par artiste, genre, album, etc.
- * support de très nombreux formats audio : **flac**, **ogg**, **mp3** mais aussi **wma**, **aac**, etc. Grossièrement, tout ce qui ne comprend pas de DRM est supporté, même sous linux ;
- * gestion de listes de morceaux (*playlist*) ;
- * protocole documenté et ouvert (et même utilisé par des produits concurrents comme le **soundbridge** de **roku**) ;
- * extensibilité par plugins ;
- * interface à l'apparence modifiable par *skins*.

Au premier démarrage, on utilise l'interface web (accessible sur le port 9000) pour préciser le dossier racine de la bibliothèque musicale (si on utilise plusieurs dossiers séparés, il faut créer un dossier maître contenant des liens symboliques vers les autres dossiers). Le logiciel parcourt alors l'ensemble de la bibliothèque pour indexer les fichiers à partir des métadonnées. Après quelques minutes (tout dépend de la taille de la base et de la puissance du serveur) la bibliothèque est accessible par le réseau.

Tout client compatible avec le protocole de **slimserver** peut naviguer dans la bibliothèque et demander à recevoir un fichier musical (la communication se fait sur le port 3483). Les fichiers sont transmis à divers formats (**flac**, **mp3**, **wma** et **PCM**), les autres formats (comme le **ogg** par exemple) étant traduits au vol par le serveur sans perte (c'est-à-dire soit en **PCM**, soit en **flac**).

En l'absence de **squeezebox**, **slimserver** reste utile : il est fourni avec un client logiciel écrit en **Java**, **softsqueeze**, qui émule le comportement de la **squeezebox**. On peut donc profiter de la bibliothèque sur un PC, à travers le réseau. En outre, **slimserver** peut envoyer sur le réseau un flux **mp3**, de la même façon qu'un serveur **SHOUTcast**. Tout client classique (**xmms**, **amarok**, etc.) peut se connecter au flux. Son contenu est contrôlé par l'interface web.

La squeezebox

Pour l'audiophile, la pièce maîtresse de notre solution est la **squeezebox**. Ce petit appareil se connecte à un réseau **ethernet** (par **wifi** ou par fils) afin de recevoir le flux musical que lui transmet le logiciel **slimserver**. La **squeezebox** possède un affichage fluorescent très lumineux et lisible, qui permet, combiné à la télécommande, de naviguer dans la bibliothèque musicale pour choisir ce qu'on souhaite écouter.

La force de la **squeezebox** est la qualité du son qu'elle délivre, depuis la version 2 (la version actuelle est la 3). La version 1 n'était pas considérée comme du matériel audiophile : la sortie analogique donnait des résultats moyens et les sorties numériques étaient desservies par une horloge bas de gamme entraînant des instabilités importantes (certains DAC refusent d'ailleurs de travailler avec un flux en provenance de cette machine). La version 2 a été conçue pour palier ces problèmes :

- * elle utilise une puce de conversion numérique/analogique du constructeur Burr-Brown très réputé dans le monde de la hifi (les puces en question équipent par exemple le matériel haute gamme de Denon) ;
- * elle utilise deux cristaux distincts pour engendrer les fréquences d'horloge du DAC et des sorties numériques. La version précédente utilisait un circuit PLL à fréquence ajustable, une solution peu coûteuse et assez classique, mais donnant de mauvais résultats. De plus un bug dans une des puces entraînait une imprécision du réglage du circuit PLL ;
- * l'alimentation à découpage, principale source de bruit dans ce genre de matériel, est isolée et filtrée au maximum. Dans la version 3, des régulateurs linéaires (moins bruités que ceux à découpage) sont utilisés pour le DAC et pour l'amplificateur (sortie ligne).

Le fondateur de **slimdevices**, Sean Adams, intervient régulièrement sur les forums consacrés à la **squeezebox**, en particulier sur le forum audiophile. L'intégration de régulateurs linéaires dans la version 3 a été fait sur les conseils d'un intervenant du forum (Andrew L. Weekes), spécialiste du design d'alimentations audiophiles. Adams a testé les régulateurs, mesuré leur impact sur la qualité du signal et a finalement décidé

de les intégrer dans la version 3 de la machine. L'ouverture de la société est assez étonnante, Adams n'ayant pas hésité par exemple à poster sur le forum le schéma électronique de la **squeezebox** pour faciliter les modifications par des tiers.

Conclusion

Notre solution permet d'obtenir un son excellent avec tous les avantages du stockage numérique de la musique. Malgré sa qualité, elle possède quelques défauts. Le problème essentiel est que la **squeezebox** n'est pas un matériel autonome. Avec une connexion internet, elle peut servir de radio numérique, car le service gratuit **squeezenetwork** remplace en partie **slimserver**. Cependant, pour accéder à une bibliothèque musicale, il faut impérativement faire tourner **slimserver** sur un PC, qui doit donc être allumé quand on veut écouter de la musique. Le coût de la solution, très faible pour le monde audiophile, reste aussi une barrière car il faut compter environ 900 à 1 000 € pour la **squeezebox** et un PC d'entrée de gamme (avec écran plat) comme serveur. Les libristes convaincus regretteront peut être que **slimdevices** ne diffuse pas les sources du firmware de la **squeezebox** (entre autres car elles contiennent du code issu du kit de développement du CPU Ubicom IP3K utilisé) et souhaite supporter à terme les fichiers à DRM (en hard).

Bien que la **squeezebox** possède des concurrents (en particulier le **Soundbridge** de **Roku**, basé sur les mêmes principes), elle semble la meilleure solution pour l'instant, en terme de qualité sonore, de rapport qualité/prix, et d'ouverture. On peut noter qu'aucune autre solution n'offre un support officiel de Linux comme OS du serveur de fichier, un argument qui suffit à lui seul pour décider le linuxien.