



Réseau des bibliothèques

AT5 CDU

Analyse technique des SIGB Koha et PMB

Benoît BARBIER

Du 18/05 au 31/08/2005

Table des matières

Introduction	i
Convention typographique	iii
1 Facilité d'installation	1
1.1 Koha sous Linux	2
1.1.1 Historique	2
1.1.2 Choix de la distribution	2
1.1.3 Installation du système de base	5
1.1.3.1 Se procurer le(s) cdrom	5
1.1.3.2 Graver les CD	6
1.1.3.3 Installation proprement dite	6
1.1.4 Installation des logiciels nécessaires au fonctionnement de Koha	6
1.1.4.1 Solution proposée pour installer les logiciels de base	7
1.1.5 Installation de Koha	7
1.1.5.1 Télécharger Koha	7
1.1.5.2 Installer Koha	8
1.1.6 Conclusion	10
1.1.6.1 Avantages de l'automatisation	10
1.1.6.2 Inconvénients de l'automatisation	10
1.2 Koha sous Windows	11
1.3 PMB sous Linux	12
1.3.1 Documentation sur l'installation de PMB	12
1.3.2 Facilité d'installation	14
1.4 PMB sous Windows	15
2 La robustesse	16
2.1 Architecture des applications Koha et PMB	16
2.1.1 Les étapes de la relation client/serveur	16
2.1.2 Différence entre CGI et module d'Apache	17
2.1.2.1 Par un module d'Apache	17
2.1.2.2 En tant que script CGI	17
2.1.2.3 Conséquences	18
2.1.3 Stabilité d'une application Web du côté du client	18
2.1.4 Stabilité d'une application Web du côté du serveur	18
2.2 Koha sous Linux	19
2.2.1 Stabilité de Koha	19
2.2.2 Stabilité de Koha du côté du navigateur (client)	19
2.2.3 Performances et matériel requis	20

2.3	Le modèle de développement de Koha	22
2.3.1	Le design de l'interface de koha.	22
2.3.1.1	Les balises de templates	23
2.3.2	Le développement par modules.	23
2.3.3	La répartition des composants	24
2.3.3.1	Les paquetages	24
2.3.4	Suggestions sur le développement de Koha	25
2.3.5	Conclusion :	25
2.4	Efficacité des fonctions	26
2.4.1	Catalogage :	26
2.4.2	Circulation :	27
2.4.3	La gestion des utilisateurs	27
2.4.4	Le client Z3950	27
2.4.4.1	Le script d'initialisation du démon Z3950	27
2.4.4.2	Utilisation du client z3950	28
2.5	Structure des données	28
2.5.1	La structure des données de Koha	28
2.5.1.1	Le catalogage non MARC	29
2.5.1.2	Le catalogage au format MARC	29
2.5.1.3	L'acquisition	30
2.5.1.4	La circulation	30
2.5.2	La structure des données de PMB	30
2.6	Koha, PMB et leurs SGBD	30
2.7	Le code de PMB	31
3	Sécurité	33
3.1	Koha sous Linux	33
3.1.1	Sécurité de l'application	33
3.1.1.1	Stockage des mots de passe	33
3.1.1.2	Habilitations des utilisateurs	34
3.1.2	Sauvegarde des données	34
3.1.2.1	Les possibilités de sauvegardes propres à Mysql	35
3.1.2.2	Mise à disposition des données par un autre logiciel	40
3.1.2.3	Les possibilités de sauvegardes fournies avec Koha	41
3.1.2.4	Politique de sauvegarde	42
3.1.3	La sécurité du serveur http	42
3.1.3.1	Identité sous laquelle est démarré le serveur	42
3.1.3.2	Répertoire racine du serveur	43
3.1.3.3	Accès à partir de certains domaines uniquement	43
3.1.4	la sécurité du serveur	44
3.2	Koha sous Windows	45
3.3	PMB sous Linux	46
3.4	PMB sous Windows	47
4	Import/export des données	48
4.1	Facilité d'importation des notices dans tous les formats standards (MARC, ISO, etc)	48
4.1.1	Importation des données provenant de CDS/ISIS	49

4.2	Facilité d'exportation des notices dans tous les formats standards (MARC, ISO, etc)	51
5	Qualité des aides	52
5.1	Le site de la documentation officielle de Koha;	52
5.1.1	Conclusion :	54
5.2	Les listes de diffusion	54
5.3	Aide en ligne	54
5.4	La documentation de PMB	55
6	Possibilité d'évolution	56
6.1	Évolutions à court terme	56
6.1.1	Gestion multi-site améliorée	57
6.1.1.1	Budget	57
6.1.1.2	Site distinct	57
6.1.2	Autres améliorations	57
6.1.2.1	Tracer qui fait quoi	57
6.1.2.2	Commandes en retard	57
6.1.2.3	Traceur de l'historique du libraire	58
6.1.2.4	Liste de circulation	58
6.1.2.5	Alerte de parution d'un périodique	58
6.1.2.6	Calcul de la date de retour	58
6.1.3	Conclusion	58
6.2	Possibilité d'ajout de fonctionnalités à Koha	58
6.2.1	Mise à jour suite à une modification du code source	59
6.2.2	Conclusion	60
6.3	Possibilité d'évolution de PMB	60
7	Synthèse et conclusion	62
7.1	Le logiciel libre	62
7.2	Facilité d'installation	64
7.2.1	L'assistant d'installation de Koha	65
7.2.2	L'assistant d'installation de PMB	65
7.3	Facilité d'utilisation	66
7.3.1	PMB	66
7.3.2	Koha	66
7.4	Robustesse et Sécurité	67
7.4.1	Koha	67
7.4.2	PMB	67
7.5	Import/export des données	67
7.6	Les aides	68
7.6.1	Installation	68
7.6.2	Utilisation	68
7.7	Pérennité et possibilité d'évolution	68
7.8	Conclusion	69

TABLE DES MATIÈRES	5
A Description de l'installation de Koha	71
A.1 Télécharger koha	72
A.2 Installer Koha	73
A.2.1 Exécuter le script d'installation	73
A.2.2 L'assistant d'installation de Koha	73
B Le serveur Web Apache	80
B.1 Configuration du serveur Apache	80
B.1.1 Les hôtes virtuels	80
B.1.1.1 Détail des directives	81
B.1.2 Modification du fichier httpd.conf	82
C Le SGDB MySQL	83
C.1 Sauvegardes des données	83
C.1.1 Copie avec tar	83
C.1.2 mysqlhotcopy	83
C.1.3 mysqldump	84
C.1.4 SELECT	85
C.1.5 Mise en oeuvre d'une réplication des serveurs	86
C.1.5.1 Règles et erreurs à ne pas commettre lors de la réplication avec MySQL	86
C.1.5.2 Opération à accomplir sur le serveur maître :	87
C.1.5.3 Opération à accomplir sur le serveur esclave :	88
D Debian GNU/Linux	91
D.1 Installation de la Debian Sarge	91
D.2 Installation des logiciels de base	91
D.2.1 Liste des logiciels à installer	91
D.2.2 Prérequis	92
D.2.2.1 Méthode de connexion	92
D.2.3 Installation de Perl	93
D.2.3.1 Installer Perl et les librairies disponibles en paquet Debian :	93
D.2.3.2 Installer les autres librairies avec l'utilitaire "Perl CPAN" :	93
D.2.3.3 La boîte à outils Yaz	96
D.2.3.4 Terminer l'installation des modules Perl	97
D.2.4 Installation et configuration de MySQL	97
D.2.5 Installation et configuration d'Apache	98
D.3 Installation du système de base	98
D.4 La gestion différée des tâches	98
D.4.1 Le gestionnaire at	98
D.4.2 Le gestionnaire cron	99
D.4.2.1 description détaillée de la syntaxe d'une crontab	99
D.4.2.2 Exemples :	100
D.4.3 Références	100
D.5 L'éditeur de texte vim	100

E	Annexes diverses	102
E.1	Définitions	102
E.1.1	Image ISO	102
E.1.2	Miroir	102
E.1.3	URL	102
E.1.4	Règles de normalisation	103
E.1.5	Le modèle relationnel	103
E.1.6	Méta données	104
E.1.6.1	Exemple d'application simplifié	104
E.1.6.2	Principe relationnel des méta données	105
E.1.6.3	Les méta données dans Koha	106
E.1.7	Clé primaire	107
E.1.8	Chemin UNIX	108
E.2	Trucs et astuces	108
E.2.1	Monter une image ISO en loopback	108
E.2.2	Choisir un bon mot de passe	108
E.3	Notes destinées aux informaticiens	109
E.3.1	La table MARC_BIBLIO	109
E.3.2	L'usage et la localisation des modules dans Koha	109
E.3.3	Programme d'importation de données provenant de CDS/ISIS.	109
E.3.3.1	Importation des données provenant d'un fichier XML avec l'API SAX.	110

Introduction

La CUD (Commission Universitaire pour le Développement) est un organisme chargé, au sein du CIUF (Conseil Interuniversitaire de la Communauté Française de Belgique), de la mise en oeuvre de la politique de coopération universitaire au développement. Elle est un lieu de dialogue et de concertation. Elle s'efforce de mettre en commun les ressources et les potentialités des universités francophones de Belgique pour augmenter ainsi l'efficacité de leur contribution à la coopération internationale et rendre possible la réalisation de projets qu'aucune institution n'aurait la capacité de réaliser seule.

La CUD, dans le cadre de ses projets de coopération, a lancé le projet AT5. Celui-ci est destiné à analyser les SIGB¹ open source qui seraient les mieux appropriés pour la gestion des bibliothèques des pays partenaires de la CUD.

Deux groupes d'analyse ont été créés. Le premier comprend des étudiants pour le diplôme de conservateur de bibliothèque de l'Enssib. Ils sont chargés de l'analyse bibliothéconomique des SIGB open source existants.

Cette recherche leur a permis de repérer 24 de ces logiciels qu'ils ont analysés.

Une première analyse a été réalisée selon un canevas d'évaluation élaboré en collaboration avec le GTRD (groupe de travail en ressources documentaires) et les stagiaires AT2 (stagiaires en formation bibliothèque).

Suite à cette analyse, ils ont retenu un seul logiciel correspondant aux exigences bibliothéconomiques des partenaires (Koha) qu'ils ont étudié en détail.

Ils ont aussi souligné les qualités de PMB qu'ils ont cependant rejeté car il ne correspond pas à un certain nombre d'exigences.

J'ai été engagé à l'Université de Liège pour effectuer l'analyse technique de Koha. Le SIGB PMB ayant d'indéniables qualités, il m'a paru pertinent d'en faire l'étude sous les aspects suivants :

- La facilité d'installation sous GNU/Linux et sous Windows
- La robustesse
- La structure des données
- La sécurité

L'objectif du présent rapport est de présenter l'évaluation de Koha du point de vue informatique et de déterminer si il est approprié ou non pour la gestion des bibliothèques universitaires des pays partenaires de la CUD.

¹ Système Intégré de Gestion de Bibliothèque

Ce rapport se présente de la manière suivante :

- L'évaluation, à travers une grille, de différents aspects techniques tels que la facilité d'installation, la stabilité, les performances, la sécurité, l'efficacité réelle des fonctions, l'interopérabilité, la structure des données, la possibilité et la facilité d'ajout de modules, la validité des aides, le paramétrage et le portage des données ;
- Les annexes techniques.

Ces annexes constituent un manuel thématique dans lequel chaque commande et son résultat sont décrits.

Son objectif est de permettre à tout un chacun d'installer un système d'exploitation GNU/Linux, Koha et les quelques logiciels nécessaires à son fonctionnement, sans pour autant disposer de compétences informatiques particulières.

Les informaticiens y trouveront aussi de nombreux détails techniques. Ces manuels leur proposent des ébauches d'études pour le développement de fonctionnalités utiles, telles que l'importation de données provenant de SIGB comme CDS/ISIS (voir paragraphe [E.3.3](#)), et quelques pistes comme par exemple la description détaillée de la mise en place d'une réplication des bases de données (voir paragraphe [C.1.5](#)).

Ce manuel peut être divisé en cinq parties :

- A. Description de l'installation de KOHA :** ce chapitre décrit toutes les commandes à employer pour télécharger KOHA et déposer l'archive sur un serveur. Il présente également les interactions possibles de l'utilisateur avec l'assistant d'installation de KOHA ;
- B. Le serveur Web Apache :** ce chapitre explique quelles sont les directives à utiliser pour configurer des hôtes virtuels et donne une explication détaillée de leur fonctionnement tout en proposant quelques exemples d'utilisations. De plus, quelques mesures de sécurité sont indiquées afin d'éviter les principales erreurs de configuration.
- C. Le système de gestion de base de données MySQL :** cette partie décrit les possibilités de sauvegarde des données avec MySQL ainsi que la mise en place d'une réplication de serveurs.
- D. Debian GNU/Linux :** ce chapitre présente l'installation du système, des logiciels de base ainsi que chaque étape de la configuration et la compilation des modules Perl. Il s'y trouve aussi des instructions détaillées sur l'installation et la configuration de MySQL, l'installation et la configuration d'Apache.
- E. Divers :** cette partie comprend un glossaire de termes et concepts informatiques.

Convention typographique

1 Commandes

Pour l'ensemble de ce document, le texte en caractère d'imprimerie désigne ce qui ne fait pas partie du contenu textuel. Lorsque vous rencontrerez cette typographie vous saurez qu'il s'agit d'une commande ou du résultat de celle-ci à l'écran. Cette typographie peut aussi servir à désigner le chemin dans l'arborescence des répertoires vers un fichier, comme par exemple :

```
/répertoire/sous-répertoire/fichier
```

1.1 Interpréteur de commande

Un shell est l'interpréteur de commande du système d'exploitation.

L'interpréteur de commande de Windows est couramment appelé le DOS.

Pour les autres systèmes tels que MacOSX, BSD et Linux, il est possible de changer d'interpréteur de commande². Dans ce document, ils seront désignés sous le terme générique de "shell".

1.1.2 Prompt

Chaque interpréteur de commande dispose d'un "prompt". Sous DOS il est représenté par :
C: \

Sur les autres systèmes, il existe une convention qui distingue le prompt de l'administrateur du système de celui des autres utilisateurs sans droit d'administration.

- sh# le prompt se termine par un # quand la commande doit être entrée en tant qu'administrateur du système d'exploitation³.
- un sh\$ le prompt se termine par un \$ quand la commande peut être entrée sous l'identifiant d'un simple utilisateur du système sans droit d'administration.

Si une commande est entrée sur plusieurs lignes dans un shell, les lignes qui suivent la première commencent par : >

²La plupart du temps on laissera le bash qui est l'interpréteur de commande par défaut. C'est aussi celui qui sera utilisé dans le cadre de cette analyse, car c'est le plus convivial et le mieux adapté aux tâches d'administration courantes.

³L'administrateur des systèmes Unix tels que Linux est couramment appelé "root".

Il est ainsi possible d'effectuer des retours à la ligne, l'interpréteur comprend que l'entièreté de la commande n'a pas été tapée.

exemple :

```
sh$ for files in /tmp/*.tar.gz
> do tar -xvzf $files
> done
```

Dès que le dernier mot de la syntaxe (le "done" d'une itération dans cette exemple) aura été entré, le retour à la ligne lance l'exécution de la commande.

1.2 Console SQL

Une console SQL est l'interface qui permet d'interagir avec un serveur de base de données. Par convention, dans ce document, si une commande est entrée dans une console SQL, la ligne commence par :

```
sql>
```

Si une commande est entrée sur plusieurs lignes dans une console sql, les lignes qui suivent la première commencent par :

```
->
```

exemple :

```
sql> SELECT * INTO OUTFILE '/tmp/testbebackup.bkp'
-> FROM borrowers where userid='un_login';
```

2 Exemple

Dans tous les cas, les commandes, ainsi que leurs résultats éventuels, sont affichés dans une police fixe. Voici par exemple une commande affichant le contenu du répertoire temporaire du système et son résultat.

```
sh$ ls /tmp
CecileGass_rapport_MigrationPMB.rtf  ManualIsisMarc153.zip  testbebackup.bkp
esrv1000-debcelem=                   marc.pl                 Winisis15_3.exe
gconfd-benoit/                       orbit-benoit/         WinIsis.doc
Guide_documentation.pdf              sessionlog             WinIsis.doc.gz
Koha.                                  SetupIsisMarc153.exe  Winisis.pdf
Koha.dump                             suisse.index.2004
Koha.tar.bz2                          sylpheed-1000=
```

3 Les notes destinées aux informaticiens

Les notes destinées à fournir un complément d'information aux informaticiens sont :

- Encadrées dans une boîte comme celle-ci si elles ne dépassent pas quelques lignes

- Placées en annexe avec l'indication de la page ou du paragraphe où elles se trouvent, si elles sont plus longues.

Ces notes ne sont pas nécessaires à la compréhension du texte.

4 La polysémie dans le vocabulaire informatique

On utilise par exemple, le mot **serveur** aussi bien pour désigner un logiciel s'exécutant sur un ordinateur comme Apache serveur Web, MySQL serveur de base de données, que pour désigner l'ordinateur lui même. Pour éviter les confusions entre l'ordinateur et le logiciel, le mot serveur sans précision désignera généralement l'ordinateur, le logiciel sera précisé par son rôle, comme serveur Web, serveur de base de données, etc.

Chapitre 1

Facilité d'installation

L'énumération de chaque étape de l'installation ainsi que les exemples ont pour but de mettre en évidence de manière objective son niveau de complexité. Le lecteur pourra ainsi confronter mes conclusions à sa propre opinion sur la difficulté des tâches à accomplir.

Cette partie de l'analyse repose sur l'installation :

- Du système d'exploitation.
- Des logiciels de base nécessaires au fonctionnement du SIGB.
- Du Système de gestion bibliographique.

Chacune de ces étapes est divisée en deux parties. La partie analyse est décrite ci-dessous et les parties techniques d'installation, de déploiement ou de configuration se trouvent en annexe.

L'installation du système d'exploitation ainsi que le choix de celui-ci sera analysé en détail pour le système GNU/Linux.

Après un bref aperçu de l'origine et de l'architecture commune à tous les systèmes GNU parmi lesquels il existe une **très grande diversité de distributions**, la justification du choix de la distribution sera développée en détail.

Le choix du système GNU/Linux repose sur les critères suivants :

- Sa pérennité sous les aspects du développement et de son cycle de vie une fois installé.
- La qualité des outils de maintenance et de mise à jour de sécurité des logiciels.
- L'importance et le dynamisme de la communauté d'utilisateurs et de développeurs.
- La fiabilité des cycles et méthodes de développement.

On terminera par énumérer les avantages et les inconvénients de la distribution choisie¹.

L'installation des logiciels nécessaires au fonctionnement de Koha est réalisable de trois manières différentes :

1. Une installation totalement manuelle telle qu'elle est décrite dans la documentation de Koha².

Cette documentation étant incomplète et obsolète, j'ai décidé de la réécrire entièrement, vous la trouverez au paragraphe **D.2**.

¹Les justificatifs de ce choix sont purement empiriques, je me base ici sur mon expérience des systèmes GNU/Linux et me contente d'expliquer pour quelles raisons j'ai choisi la distribution Debian. Il ne s'agit pas d'une sélection méthodique utilisant des critères prédéfinis. Ce serait impossible car certains de ces critères tels que le cycle de développement en 3 stades, sont propres à la Debian.

²http://www.kohadocs.org/Installing_Koha_on_Debian_Woody.html

Les étapes et les difficultés rencontrées par cette façon de faire seront décrites au paragraphe 1.1.4.

2. Une installation partiellement automatisée est proposée comme alternative à l'installation manuelle au paragraphe 1.1.4.1.
3. Enfin une installation totalement automatisée, y compris l'installation de Koha, est proposée comme alternative aux deux possibilités précédentes au paragraphe 3.

1.1 Koha sous Linux

L'installation n'est pas très aisée, elle requiert des compétences dans le domaine des SIGB (lors de la définition des formats MARC), ainsi que quelques compétences en informatique.

1.1.1 Historique

GNU est l'acronyme récursif de GNU's Not Unix. Le projet GNU a été lancé en 1984 par Richard Stallman afin de développer un système d'exploitation complet et gratuit. Le projet GNU ne disposait pas de noyau lors de sa fondation. Linus Torvalds a été le développeur principal du noyau Linux. Ce noyau occupe à peine plus de 1Mo. Linux n'est donc que le noyau de cet ensemble composé de plusieurs Go de logiciels. L'appellation Linux est abusive et tend à minimiser l'importance du précurseur de ce projet qui est Richard Stallman.

C'est bien Linux qui a contribué à l'essor du logiciel libre en attendant la première version stable du noyau GNU Hurd.

L'architecture GNU est basée sur le principe suivant : spécialiser au maximum les programmes et déléguer **les fonctions communes à plusieurs programmes** à des bibliothèques partagées.

Les caractéristiques du système d'exploitation sont :

Spécialisation : concevoir un logiciel pour une seule tâche, mais il doit s'en acquitter à la perfection.

Délégation : un logiciel GNU est généralement très léger (1 ou 2 Mo). Le principe est de s'adresser à des bibliothèques partagées spécialisées qui sont d'autres programmes installés séparément.

On économise par la même occasion l'espace mémoire car une bibliothèque est chargée en mémoire une seule fois pour plusieurs programmes qui en dépendent.

Chaque brique de ce robuste ensemble est confiée à un développeur spécialiste de la tâche qu'il s'est choisie.

1.1.2 Choix de la distribution

Debian est la distribution GNU/Linux qui a été retenue pour cette étude pour les raisons suivantes :

1. La réactivité des mises à jour de sécurité est très rapide (de quelques heures à 3 jours).

2. Son extraordinaire système de paquetage est un des modèles de paquetage le plus abouti.

J'entends, par système de paquetage, l'ensemble des outils d'installation qui permettent d'assurer la cohérence des versions, et les dépendances entre logiciels et librairies³.

Chaque programme est empaqueté dans une archive appelée paquetage ou paquet. Cette archive contient, entre autres, la liste des autres paquets dont elle dépend.

Sous Debian, le gestionnaire de paquetage s'appelle apt, il suffit de lui demander d'installer un logiciel pour qu'il vérifie la liste des dépendances et localise le paquet à installer grâce à un fichier appelé source.list.

Le système de paquetage apt propose le cas échéant une liste de paquets à installer pour satisfaire les dépendances.

La tâche de l'utilisateur de ce système se résume à deux opérations :

- (a) Demander au système de paquetage apt d'installer un paquet en tapant cette commande :

```
sh# apt-get install nom_du_paquet
```

- (b) Appuyer sur la touche "Enter" du clavier

Apt liste les paquets dépendants suivis de Y/n.

L'utilisateur valide et le travail s'accomplit automatiquement, ainsi qu'une mise à jour éventuelle du paquet cible si il était déjà installé.

Comme je l'ai indiqué plus haut, apt gère automatiquement la cohérence des versions des paquets que l'on désire installer. Il n'est donc généralement⁴ pas nécessaire de s'en préoccuper. On indique à apt le nom du programme sans préciser la version du paquet ou du programme. Cette tâche complexe revient donc au développeur Debian qui est chargé de la maintenance du paquet.

3. Le dynamisme des listes de diffusion qui comportent notamment une liste d'utilisateurs, et une liste de développeurs, toutes deux francophones.

<http://www.debian.org/MailingLists/subscribe>

4. Son modèle de développement en 3 stades (unstable, testing, stable) :

Unstable alias Sid⁵ est la version instable, celle sur laquelle travaillent les développeurs. Elle est appelée ainsi parce qu'elle risque de casser lors d'une mise à jour. Ce risque de rupture des dépendances de librairies provient du fait qu'un développeur travaillant sur une librairie peut, en accord avec l'équipe de développement, en modifier la structure. Mais rien ne garantit que les logiciels qui en dépendent seront mis à jour simultanément. (Par exemple, si un utilisateur fait une mise à jour de son système à cet instant les dépendances sont rompues.)

Unstable est en même temps un stade de test de la version testing. Si aucun bug majeur n'a été signalé dans un paquet depuis plusieurs semaines, il passe de unstable à testing. Mais en cas de problème survenant ultérieurement, il retourne en unstable.

³Des librairies sont des ensembles de sous programmes communs à plusieurs logiciels installés. Elles peuvent donc partager des fonctionnalités entre plusieurs logiciels.

⁴Il y a certaines exceptions, par exemple pour le serveur Web Apache on peut choisir entre la version 1 courante ou la version 2 courante, car le développement de ce logiciel se fait sur deux branches parallèles. Ce principe de développement parallèle de plusieurs versions se retrouve dans de nombreux logiciels tels que mysql (serveur de base de données), le noyau Linux, etc.

⁵Le garçon qui casse les jouets dans Toy Story

Testing alias etch, est le stade de test de la version stable... Il y a beaucoup de mouvements de paquets venant de unstable. C'est la version la plus appropriée pour une machine de bureau disposant d'une connexion rapide à Internet. En choisissant la version testing, on dispose de paquets récents et plus stables que ceux venant de la plupart des distributions commerciales.

Stable alias Sarge. Après une longue période de tests et de corrections de bugs, les paquets de la testing sont gelés durant plusieurs semaines pour devenir la version stable. Si un paquet cause un problème durant cette période de gel, il en est exclu. Un fois passée en stable plus aucun paquet n'est ajouté ou enlevé jusqu'à la prochaine version qui ne vient qu'après plusieurs années.

Sur cette version de la Debian, les développeurs de paquets concentrent tous leurs efforts sur la sécurité. Si une faille de sécurité est découverte dans l'un d'eux, une mise à jour est disponible dans les plus brefs délais.

C'est la version idéale pour les serveurs en production où la sécurité et la stabilité sont deux éléments importants.

Debian est le système d'exploitation choisi et proposé comme support de Koha dans le cadre de cette analyse. Cette version se prête tout particulièrement bien à l'usage qui en sera fait dans les pays du sud.

Les avantages de la version stable sont :

- Sa pérennité : On installe le système d'exploitation pour plusieurs années, il ne subit que quelques modifications mineures survenant suite à la découverte d'un trou de sécurité sur un des paquets installés.

- Sa robustesse et sa stabilité.

Les paquets de la version stable actuelle ainsi que leurs interdépendances ont subi :

- (a) Plusieurs semaines de tests avant d'être acceptés en testing (avec corrections de bugs).

- (b) Plus de 3 ans d'évaluation en testing (avec corrections de bugs)

- (c) Plusieurs mois de gel d'évaluation et de débogage⁶ intensif avant de devenir stable.

A ma connaissance, aucune autre distribution ne subit de tests aussi prolongés par une aussi grande communauté d'utilisateurs⁷.

Par exemple, lorsque Redhat (qui est une autre distribution) sort une nouvelle distribution, celle-ci n'a été testée que par les développeurs et une communauté restreinte de beta testeurs, ces tests durent tout au plus quelques mois.

- Son haut niveau de sécurité et le dynamisme des mises à jour en cas de faille.
- Sa très grande facilité de maintenance.

⁶Cette opération consiste en la correction d'erreurs dans un logiciel

⁷qui en font un usage quotidien sur leur machine de bureau durant plusieurs années

Son inconvénient :

- Son obsolescence : après un ou deux ans, un développeur qui a besoin de travailler avec des bibliothèques récentes doit passer en testing et utiliser certains paquets de unstable.

Pour les détails d'installation d'un système Debian en stable (Sarge) consultez le paragraphe D.3 en annexe.

1.1.3 Installation du système de base

J'entends par système de base un système d'exploitation disposant du strict minimum pour pouvoir fonctionner, interagir avec les périphériques écran et clavier mais aussi avec l'utilisateur, il doit donc disposer d'un interpréteur de commande. Il doit aussi être correctement configuré pour s'intégrer dans un réseau.

Pour que cette partie du travail aboutisse avec succès, certaines conditions devront être remplies. Elles seront énumérées et les termes techniques seront définis. S'il s'avère que, dans certains cas, toutes les conditions minimales ne sont pas réunies, nous évaluerons les alternatives possibles.

Ces conditions s'évaluent en terme :

- de compétences minimales que requiert cette étape ;
- d'infrastructure réseau ;
- du matériel disponible⁸ ;

1.1.3.1 Se procurer le(s) cdrom

Si l'on dispose d'un accès à Internet via une connexion rapide on peut soit :

- télécharger l'image iso (voir paragraphe 1.1.3.2) de l'installation réseau qui fait 108 Mo. En partant du principe que l'architecture de l'ordinateur sur lequel on désire installer Koha soit de type i386⁹ :

<http://www.debian.org/CD/netinst/>

Choisissez donc i386.

- télécharger les 14 CDrom d'installation. Pour ce faire, sélectionnez le miroir (voir définition au paragraphe E.1.2) le plus proche de chez vous :

<http://www.debian.org/CD/http-ftp/>

Par exemple pour la Belgique l'adresse de la page de téléchargement est la suivante :

<http://ftp.belnet.be/debian-cd/current/i386/iso-cd/>

Si l'on ne dispose pas de connexion rapide, il faut soit :

- être patient, aussi bien pour le téléchargement de l'image du CD d'installation par réseau que pour procéder à l'installation elle-même. De plus, il faut avoir la possibilité de monopoliser la bande passante d'une connexion PSTN (ligne téléphonique) durant plusieurs heures.
- se faire envoyer les 14 CD par la poste et un 15^e contenant l'archive de Koha

⁸Je ne parle pas du matériel fourni par la CUD mais de celui qui doit être présent sur place avant toute installation

⁹i386 désigne toutes les architectures PC compatible Intel. C'est à dire 386, 486, Pentium I à Pentium 4, et ses compatibles de chez AMD, Via, etc

1.1.3.2 Graver les CD

Cette opération exige :

- De savoir ce qu'est une image iso.
Une image iso est un fichier unique qui contient l'ensemble du système de fichier (au format iso9660) d'un cdrom ou d'un dvd. C'est ce fichier qu'il faut télécharger et graver sur le cdrom.
- Qu'un graveur de cdrom ou de dvd soit installé sur un ordinateur disposant d'un logiciel de gravure approprié.

1.1.3.3 Installation proprement dite

Cette opération requiert de disposer des CDrom d'installation et de quelques connaissances de base :

- Partitionner le disque et donc savoir ce qu'est une partition primaire, étendue, logique. Bien qu'il soit possible de choisir des tables de partition prédéfinies dans un menu.
- Savoir ce qu'est un secteur d'amorçage pour pouvoir installer le gestionnaire de démarrage du système.
- Choisir la langue, le clavier, le fuseau horaire.
- Comprendre la distinction entre le mot de passe root et le mot de passe utilisateur.
- Configurer à l'aide d'un assistant le système de paquets. Celui-ci pose des questions qui requièrent un minimum de connaissances en informatique, il faut savoir ce qu'est un site miroir, un accès par ftp ou http¹⁰.

Si ces conditions ne sont pas remplies, des solutions alternatives sont décrites au paragraphe [1.1.6](#)

1.1.4 Installation des logiciels nécessaires au fonctionnement de Koha

Une liste des logiciels nécessaires au fonctionnement de Koha se trouve en annexe au paragraphe [D.2.1](#). Cette étape de l'analyse a été réalisée via une connexion distante. La machine cliente dispose d'une interface graphique et d'un navigateur. On se base sur le principe que :

- le serveur sur lequel on veut installer Koha est une installation de base d'une Debian GNU/ Linux en version stable.
- aucun autre logiciel que ceux fournis avec l'installation de base n'est installé
- une connexion à un terminal en administrateur (locale ou distante (Cf. Paragraphe [D.2.2.1](#))) est établie.

Si la personne chargée d'installer Koha décide d'installer les logiciels de base manuellement, tout de suite après l'installation de MySQL, il est recommandé d'indiquer un mot de passe pour l'administrateur du serveur.

Dans les autres cas, par exemple si cette personne utilise le script d'automatisation de l'installation que j'ai rédigé (voir paragraphe [1.1.4.1](#)), il n'y a pas à s'en soucier mais il faudra le conserver précieusement car il lui sera demandé lors de l'installation de Koha (Cf. Paragraphe [1.1.5.2](#))¹¹

¹⁰Ces notions seront expliquées en annexe au paragraphe [D.1](#)

¹¹Il est possible de modifier ce script pour qu'un mot de passe aléatoirement généré soit proposé par défaut.

L'installation des logiciels de base comprend aussi celle des modules Perl dont Koha a besoin. C'est la moins aisée des opérations et les questions posées sont assez rébarbatives.

1.1.4.1 Solution proposée pour installer les logiciels de base

Il est indispensable d'installer quelques logiciels. Plutôt que de suivre pas à pas de nombreuses opérations et de devoir taper les commandes dans un terminal, j'ai rédigé un script qui se charge de l'installation des logiciels et modules Perl nécessaires au fonctionnement de Koha. Il sera présent sur un cdrom contenant tous les éléments nécessaires.

Dans sa version actuelle, ce script reste partiellement interactif, ce qui signifie qu'il est nécessaire, pour certaines questions, de choisir la bonne option.

Les questions demandant le plus d'attention concernent les modules Perl.

Pour éviter toutes ces questions qui concernent le téléchargement et la compilation des modules Perl, j'ai développé des paquets Debian.

Pour aider la personne qui ne désire pas utiliser mon script et mes paquets, vous trouverez en annexe au paragraphe [D.2](#) une copie d'écran de chaque question posée, un mot d'explication, et la réponse la plus appropriée selon les circonstances.

Comme indiqué au paragraphe précédent, cette solution requiert certaines connaissances en informatique. Afin de pouvoir se connecter à un terminal, il faut savoir ce qu'est un login et un mot de passe.

Si cela n'est déjà fait, il faudra également veiller à installer le système de base.

Si la personne chargée de l'installation et de la maintenance du serveur désire effectuer une partie du travail à partir de sa machine de bureau, il faut en plus qu'elle maîtrise des notions de connexion distante.

Ces notions seront aussi utiles à l'administrateur pour disposer de l'archive d'installation de Koha (Cf. paragraphe [1.1.5.1](#)).

1.1.5 Installation de Koha

1.1.5.1 Télécharger Koha

Il faut savoir qu'un serveur GNU/Linux ne dispose pas d'une interface graphique.

Une interface graphique est la couche logicielle qui permet à l'utilisateur d'interagir avec le système par des fenêtres, des boutons "cliquables" et un écran disposant d'icônes et de menus. Bref, un environnement semblable à celui de MacOS ou Windows.

Bien sûr il est tout à fait possible d'installer ces fonctionnalités sur un serveur, mais c'est **fortement déconseillé pour des raisons de sécurité**. Nous avons vu au paragraphe [1.1.1](#) que chaque programme installé fonctionne grâce à des bibliothèques partagées. Un pirate désireux de s'infiltrer dans un système peut prendre le contrôle de celui-ci en s'aidant de ces programmes. Moins il dispose de ces bibliothèques partagées, plus il aura de difficultés à le faire fonctionner.

Naturellement, il ne faut pas oublier qu'installer un grand nombre de programmes augmente

aussi le risque que l'un d'entre eux comporte une faille de sécurité exploitable.

Afin de limiter les risques, l'installateur de Koha devra télécharger l'archive d'installation dans un terminal en mode texte¹².

Pour y parvenir, il peut soit :

- Télécharger Koha à partir de son ordinateur de bureau avec son navigateur habituel, et ensuite copier l'archive sur le serveur ;
- Toujours à partir de son ordinateur de bureau, rechercher l'URL (voir définition en annexe au paragraphe E.1.3) pointant vers l'archive de Koha¹³ et ensuite copier ce lien et le coller dans une console donnant accès à une connexion distante ;
- Télécharger l'archive directement à partir du serveur dans un navigateur internet en mode texte.

1.1.5.2 Installer Koha

Cette partie du travail implique que l'installation du système d'exploitation et des logiciels de base s'est bien déroulée. Un assistant d'installation en mode texte est fourni dans l'archive de Koha.

Utiliser l'assistant d'installation ne requiert pas de compétences particulières en informatique. Il suffit de :

1. Comprendre le principe des chemins UNIX (la définition du chemin UNIX se trouve au paragraphe E.1.8 en annexe).
2. Fournir le mot de passe administrateur de la base de données MySQL.
3. Comprendre la différence entre un fichier et un répertoire, en effet l'assistant demande le répertoire dans lequel il doit installer l'OPAC et l'intranet.
4. Fournir un nom d'hôte qui se présente sous la forme :

`machine.sousdomaine.domaine.net` (exemple : `koha-admin.lib.ulg.ac.be`)

Un nom d'hôte complet permet d'identifier un ordinateur de manière unique dans un réseau.

Si vous désirez que le nom d'hôte soit différent pour l'OPAC et l'interface bibliothécaire, les étapes de définition de ce nom seront plus complexes. En effet, pour que ces noms soient reconnus sur le réseau, il faut qu'une correspondance soit faite entre ce nom alphabétique et un identifiant numérique appelé **adresse IP** qui est utilisée par les ordinateurs pour se reconnaître entre eux. Ce point dépassant le cadre de cette étude, il ne sera pas développé.

L'administrateur du réseau des bibliothèques devra créer ces noms d'hôtes.

Pour fournir un nom d'hôte, deux étapes sont nécessaires :

- (a) Obtenir de l'administrateur du réseau deux noms de domaine.
- (b) Créer des hôtes virtuels sur le serveur qui héberge Koha.

¹²Un peu comme le mode DOS mais en infiniment plus puissant, stable et confortable (avec historique des commandes, auto-complétion, copier-coller, recherche avant-arrière d'une ligne de commande avec mot-clé dans un historique), rien de comparable, si ce n'est que ce soit du mode texte

¹³Des instructions détaillées sur les opérations à accomplir sont fournies en annexe au paragraphe A.1

Des explications sur la manière de configurer un hôte virtuel sont fournies en annexe au paragraphe B.1.1.

Si aucun nom d'hôte spécifique n'a été créé par l'administrateur du réseau, il faut utiliser le nom d'hôte du serveur sur lequel on installe Koha.

5. Fournir un numéro de port

Un nom d'hôte permet d'identifier de manière unique un ordinateur dans un réseau, mais cet identifiant unique n'est pas suffisant car plusieurs services peuvent s'exécuter sur cet ordinateur.

Par exemple le navigateur internet doit-il se connecter à l'OPAC ou à l'interface bibliothécaire sachant que ces deux services sont exécutés sur le même ordinateur ?

Ce qui permet de différencier les services est un **numéro de port**. Celui-ci permet au navigateur de s'adresser au bon service sur la bonne machine grâce à la combinaison de ces deux identifiants, nom d'hôte et numéro de port.

Le numéro de port proposé par défaut par le script d'installation de Koha est le 8080. Ce numéro étant utilisé comme valeur par défaut par de nombreux services comme Tomcat, les Web cache¹⁴, il m'a semblé préférable d'utiliser le port 8082 car il n'est pas affecté à un service. Ce numéro étant libre, c'est ce port qui sera utilisé dans la suite de ce document pour identifier l'interface bibliothécaire. Bien sûr, rien ne vous empêche de laisser le choix par défaut, à condition qu'il ne soit pas utilisé par un autre service sur le même ordinateur.

L'assistant d'installation pose ensuite des questions d'ordre principalement bibliothéconomique notamment au niveau de la configuration du format des notices. Le bibliothécaire peut accomplir seul ce travail s'il a connaissance des informations énoncées dans les paragraphes précédents.

Les questions posées par l'assistant d'installation concernant les aspects informatiques proposent comme réponse une valeur par défaut qui convient.

Les seules modifications souhaitables sont :

- Le numéro du port de l'interface bibliothécaire ;
- L'installation du répertoire des log dans /var/log pour mieux respecter les standards.

1.1.5.2.1 En conclusion : l'assistant d'installation convient parfaitement aux bibliothécaires pour cette étape, mais il requiert des compétences en informatique, il est donc souhaitable que l'installation de Koha soit effectuée par un informaticien en présence du bibliothécaire. Il n'est pas possible d'éviter cette étape en l'automatisant car le bibliothécaire doit, entre autre, choisir quel format du MARC il va utiliser.

Cette décision doit être prise au préalable, l'assistant d'installation offre le choix entre le MARC21 et l'UNIMARC. Quant à l'informaticien, il est souhaitable que l'administrateur du réseau lui ait attribué un nom de domaine tel que <http://koha-admin.organisation.org> pour l'interface bibliothécaire et <http://opac.organisation.org> pour l'OPAC. Ce sera à lui d'indiquer ces noms de domaine lorsqu'ils lui seront demandés par l'assistant d'installation.

¹⁴Ces noms sont donnés à titre d'exemple. Ne pas savoir ce que c'est ne nuit pas à la compréhension du texte

1.1.6 Conclusion

Les principales difficultés rencontrées par l'installateur de Koha sont les suivantes :

- **Se procurer les images iso d'une Debian et les graver ;**
- **Installer le système de base ;**
- **Configurer le système de base** (choisir les miroirs, etc.) ;
- Entrer les commandes au clavier pour **installer les logiciels de base** ou exécuter le script ;
- **Télécharger l'archive de Koha** sur une machine de bureau pour la copier sur le serveur ou la télécharger directement à partir de celui-ci.

Il existe plusieurs solutions selon les compétences en informatique et les moyens techniques mis à la disposition du bibliothécaire ou de l'installateur de Koha :

1. **Une installation manuelle :**

Effectuer l'installation à partir des CDrom téléchargés sur le site de Debian, mettre en place les logiciels de base ((avec apt et Perl CPAN) ou utiliser le script), télécharger Koha et l'installer.

2. **Une installation partiellement automatisée :**

Après avoir largement testé et analysé la première option, j'en conclus qu'il serait possible d'automatiser une grande partie de cette tâche. Pour faciliter l'installation, centraliser les fichiers qu'il faudrait télécharger et éviter de répondre au questionnaire du script d'installation des modules Perl, le plus simple serait de créer un cdrom contenant les éléments nécessaires et grâce auquel l'installation des modules Perl serait automatisée.

Les opérations à effectuer seraient les suivantes :

Installer le système de base, configurer le clavier, partitionner, déterminer la zone horaire, etc...

Changer de CD après que le système de base ait été installé avec succès.

Lancer le script permettant d'automatiser l'installation des librairies et logiciels nécessaires au fonctionnement de Koha. Ce script se chargerait de lancer l'assistant d'installation de Koha.

3. **Une installation entièrement automatique.**

En démarrant avec une disquette ou une clef USB contenant toutes les informations nécessaires au processus d'installation.

Il est aussi possible de développer un cdrom d'installation à l'aide d'un environnement tel que fai (Fully Automatic Installation).

Voir <http://www.informatik.uni-koeln.de/fai/fai-guide.html/>

1.1.6.1 Avantages de l'automatisation

1. Elle économise la bande passante en téléchargeant voire même en s'échangeant des cdrom contenant uniquement les éléments nécessaires à la mise en place de Koha.
2. Elle facilite l'installation.

1.1.6.2 Inconvénients de l'automatisation

1. Tous les téléchargements se feront dans ce cas à partir d'un serveur de l'Université de Liège et au détriment de sa bande passante. D'où l'utilité des miroirs régionaux répartis par zones géographiques (voir paragraphe [E.1.2](#)).

2. Cette option m'oblige à mettre des images de cdrom à jour lors de la sortie de nouvelles versions de Koha ou lors d'une mise à jour de sécurité du système Debian.
3. Elle crée une dépendance de l'utilisateur et occulte une partie des compétences nécessaires à une mise en place manuelle.

Il y a donc deux formes de dépendance de l'utilisateur induites par ce choix. Mon contrat à durée déterminée parvenant à son terme, si je n'assume pas bénévolement la maintenance de ces cdrom, ils sombreront rapidement (moins de trois mois) dans l'obsolescence.

1.2 Koha sous Windows

J'ai testé et analysé l'installation de Koha sous GNU/Linux, car c'est l'environnement pour lequel a été développé Koha.

C'est aussi le système de prédilection des serveurs Web, 80% de ceux-ci tournent sur des systèmes de ce type, que ce soit GNU/Linux, BSD, ou une des nombreuses variantes des systèmes de type UNIX.

Bien qu'il existe une version de Koha pour Microsoft Windows, je ne l'ai pas testée, car c'est une adaptation de Koha vers Windows et non pas son "environnement naturel".

Il en va de même pour le serveur Apache qui a subi un portage vers Windows bien qu'il n'ait pas été initialement développé pour cet environnement.

http://apache.be.proserve.nl/httpd/binaries/win32/apache_2.0.54-win32-x86-no_ssl.msi

Il s'agit donc dans tous les cas de portages, mais pas d'un développement initial pour cet environnement.

Windows est aussi un système d'exploitation, qu'il me semble raisonnable de réserver à un usage ludique (environnement de prédilection des jeux vidéo) ou domestique, voire bureautique (dans le cas où la fiabilité des systèmes est moins importante que le respect des habitudes et conditionnements de ses usagers). Et ce, pour sa sensibilité excessive aux virus, ses nombreuses failles de sécurité, sa lourdeur en tant que serveur, et donc sa consommation excessive en ressources pour un service donné. Le risque de rupture de service me paraît aussi trop important.

Il y a aussi le coût des licences qu'il me paraît superflu d'ajouter à celui du matériel, **un des critères de sélection de Koha étant la gratuité**, cette option ne peut être retenue.

1.3 PMB sous Linux

1.3.1 Documentation sur l'installation de PMB

La documentation générale de PMB est divisée en trois parties décrites au paragraphe 5.4 du chapitre 5.

Le premier grand point positif c'est l'existence même de cette documentation qui accompagne le lecteur dans chacune des étapes de l'installation. C'est donc plein d'enthousiasme et de confiance qu'on entame la lecture de ce document. Le premier problème se pose au point 3 du chapitre 2. Configurer son serveur avant d'installer PMB :

<http://www.sigb.net/doc/html-install/ch02s03.html>

Cette partie du document est uniquement orientée windows.

Vous pouvez donc passer à la page destinée à l'installation sous Debian¹⁵.

<http://www.sigb.net/doc/html-install/ch06s03.html>

Le document pdf recommandé au début de cette page concerne la Debian Woody, il est partiellement obsolète.

Il est nécessaire qu'il soit mis à jour, sans quoi il n'est utilisable que par une personne ayant une très bonne connaissance de cette distribution.

Pour satisfaire les pré-requis du paragraphe 3.1 de cette page, il faut que les opérations suivantes aient été accomplies :

Il faut installer les paquets suivants avec la commande apt-get et l'option install :

```
sh# apt-get install libmysqlclient mysql-client-4.1 mysql-common-4.1
sh# apt-get install mysql-server-4.1 Apache Apache-Common
sh# apt-get install php4-gd php4-xslt libyaz libyaz-dev
sh# apt-get install yaz yaz-doc libapache-mod-php4 php4
sh# apt-get install php4-common php4-domxml php4-domxml php4-mysql
sh# apt-get install php4-odbc php4-pear php4-pgsql.
```

Donner un mot de passe administrateur au serveur de base de données :

```
sh# mysqladmin -u root password 'mot_de_passe_root'
```

Passons ensuite à l'installation proprement dite :

<http://www.sigb.net/doc/html-install/ch03s02.html>

Impossible de trouver le fichier "PMB.tar.gz"... Soit utilisons l'archive windows puisque les deux archives contiennent les mêmes fichiers.

Mais alors vous ne pourrez plus suivre les instructions suivantes :

<http://www.sigb.net/doc/html-install/ch06s03.html>

Il faudra improviser.

Il ne m'a pas été possible de suivre cette documentation pas à pas pour les raisons suivantes :

<http://www.sigb.net/doc/html-install/ch06s03.html>

¹⁵Debian est une distribution GNU/Linux. C'est à dire, une mise en paquets cohérente des éléments logiciels qui respecte une méthode spécifique à chacune de ces distributions.

- La documentation part du principe que le lecteur dispose d'un serveur LAMP (Linux, Apache, MySQL, PHP) opérationnel. Or sur l'ordinateur de test qui a servi à cette étude, php4 ainsi que les bibliothèques n'étaient pas installés.
- je n'ai pas trouvé l'archive tar.gz et ai utilisé l'archive pour windows compatible pour Linux et l'ai décompressée avec la commande unzip¹⁶.
- j'ai préféré laisser l'assistant d'installation créer la base de données comme indiqué à cette page :
<http://www.sigb.net/doc/html-install/ch03s02.html>

Une fois ces difficultés surmontées, en improvisant¹⁷ puis en retournant à la documentation, l'assistant d'installation est d'une convivialité remarquable. La procédure d'installation et les instructions de configuration décrites aux pages suivantes se sont déroulées avec une très grande aisance.

<http://www.sigb.net/doc/html-install/ch03s02.html>

<http://www.sigb.net/doc/html-install/ch03s03.html>

La documentation doit être corrigée sur les points suivants :

- Donner le lien vers l'archive de PMB pour Linux ou bien adapter la documentation pour décrire l'usage de l'archive Windows qui est de toute façon compatible ;
- réécrire la documentation spécifique à Debian fournie avec l'archive pour la mettre à jour en sarge ;
- Indiquer au lecteur qui suit la documentation séquentiellement de ne pas s'inquiéter si les instructions données à cette page ne correspondent pas à son système. La configuration de php.ini et de my.ini est décrite à cette page :

<http://www.sigb.net/doc/html-install/ch02s03.html>

Mais aucune indication n'est donnée sur ces points à la page spécifique au système utilisé.

ex : <http://www.sigb.net/doc/html-install/ch06s03.html>

Conséquence :

Le lecteur ne trouve pas le php.ini de son système, il doit alors accomplir les opérations suivantes :

Taper les commandes suivantes en étant logué en tant que root (administrateur du système) :

```
sh# updatedb
sh# locate php.ini
/usr/share/doc/php4-common/examples/php.ini-recommended
/usr/share/doc/php4-common/examples/php.ini-paranoid
/usr/share/doc/php4-common/examples/php.ini-dist
/usr/share/php4/php.ini-dist
/etc/php4/apache/php.ini
/etc/php4/cli/php.ini
/etc/php4/cgi/php.ini
```

La commande locate donne 7 résultats. Le cinquième est le bon /etc/php4/apache/php.ini, car apache est le serveur Web qui sera utilisé et c'est ce php.ini qui configure Apache.

¹⁶Il faudra l'installer avec apt si elle n'est pas disponible.

¹⁷Je ne dispose plus du temps nécessaire à la description des opérations que j'ai effectuées pour accomplir l'installation

Même opération pour le fichier my.ini qui sous Linux s'appelle my.cnf.

La documentation parle de fichiers aux extensions .dll. Le lecteur doit comprendre que, sous Linux, les extensions des bibliothèques partagées sont des .so.

Rien n'indique si les opérations décrites à cette page sont utiles sous Linux. Je n'ai apporté des modifications que lorsque j'y trouvais une correspondance, j'ai uniquement accompli les opérations suivantes et PMB fonctionne.

```
max_execution_time = 120
upload_max_filesize = 16M
```

Et dans :

```
/etc/mysql/my.cnf
set-variable= max_allowed_packet=16M
```

Conclusion :

L'assistant d'installation utilise une interface Web, il est auto documenté et très facile à utiliser.

La documentation dans son état actuel laisse quelques doutes qui ne risquent pas de poser de gros problèmes aux spécialistes des systèmes UNIX. Il en va tout autrement pour la personne qui voudrait se donner la peine d'installer PMB sous GNU/Linux sans en être un spécialiste. Dans ce cas, cette personne ne surmontera probablement pas certaines difficultés.

1.3.2 Facilité d'installation

Malgré les problèmes concernant la documentation, qui laisse planer quelques doutes et une fois ces problèmes surmontés, on peut vraiment dire que l'usage de l'assistant d'installation est d'une aisance remarquable. Il suffit d'installer quelques logiciels, de décompresser l'archive dans le répertoire approprié du serveur Web, d'entrer le bon URL dans un navigateur pour exécuter l'assistant d'installation.

Solution :

Faire deux paquets pour les distributions principales :

- un rpm pour les distributions basées sur rpm ;
- un Debian pour les distributions basées sur apt.

Ces paquets devraient gérer les dépendances avec un système LAMP (Linux, Apache, MySQL, PHP) et toutes les bibliothèques nécessaires à PMB.

La configuration minimale d'Apache et MySQL est de toute façon assurée par leur paquets respectifs.

Il suffirait de très peu d'interactions entre le script de configuration du paquet et l'utilisateur

- pour créer le mot de passe root (administrateur) de MySQL ;
- le compte utilisé par l'assistant d'installation de pmb ;
- et enfin un dernier message indiquant l'URL que doit utiliser la personne pour terminer l'installation avec l'assistant d'installation utilisant une interface Web.

Conclusion :

Moyennant quelques petits efforts de développement de paquets, l'installation de PMB serait d'une facilité remarquable.

1.4 PMB sous Windows

PMB est très bien documenté en ce qui concerne l'installation de PMB sous Windows.

Le paragraphe [3.2](#) consacré à Koha explique pour quelles raisons cette option n'a pas été retenue ni testée.

Chapitre 2

La robustesse

2.1 Architecture des applications Koha et PMB

Il m'a paru intéressant de rappeler les principes d'un service Web, tel que celui utilisé par Koha et PMB.

Les lecteurs maîtrisant ces notions peuvent directement passer au paragraphe [2.2](#)

Koha et PMB sont des applications Web, elles fonctionnent selon le principe de client/serveur. Ces deux SIGB ont besoin d'un serveur Web pour pouvoir fonctionner du côté serveur et d'un navigateur du côté client.

Perl et PHP sont deux langages de programmation interprétés. Ce qui signifie que leur code source est directement compréhensible par l'ordinateur moyennant une interprétation à la volée. Dans ce cas, le code est appelé un script par opposition aux codes, qui ont besoin d'être compilés¹ pour être compréhensibles par un ordinateur.

Les codes de Koha sont des scripts Perl qui sont interprétés par un interpréteur Perl lancé par le serveur Web Apache. On appelle cela du CGI (Common Gateway Interface) car le serveur Web Apache délègue au programme Perl l'interprétation du script.

Par contre les codes de PMB sont des scripts PHP interprétés directement par un module du serveur Web Apache. Dans ce cas c'est Apache qui se charge de rendre le script PHP directement compréhensible par l'ordinateur sans déléguer cette tâche à un autre programme. C'est ce qui le différencie du CGI. Cette distinction sera abordée plus en détail au paragraphe [2.1.2](#).

2.1.1 Les étapes de la relation client/serveur

1. L'internaute ouvre son navigateur² et tape l'URL ayant la forme suivante :
<http://koha-admin.domaine.be:8082/cgi-bin/koha/mainpage.pl>
Pour plus d'informations concernant la notion d'URL, consultez le paragraphe [E.1.3](#)
Le client demande au serveur le fichier "mainpage.pl".

¹La compilation est l'action de transformer un code source, fait de phrases et de mots alphanumériques, respectant la grammaire du langage de programmation et compréhensible par un programmeur, en un code binaire utilisable par un ordinateur.

²Le navigateur est un client http

Il s'agit, dans ce cas, d'un fichier explicitement mentionné dans la configuration du serveur comme étant un script.

2. Le serveur reçoit la requête du client et exécute/interprète le fichier demandé.
3. La plupart du temps ce fichier renvoie une page html en fin d'exécution.
4. Cette page html est reçue par l'application cliente (le navigateur dans ce cas).
L'internaute peut, par exemple, encoder le mot-clé recherché dans un champ du formulaire.
5. Ensuite il envoie le formulaire au serveur en validant.
6. Le formulaire comporte un fichier cible qui est chargé d'effectuer la recherche dans une base de données du serveur par exemple.
Une page html affichant le résultat de la recherche est générée par le script et envoyée au client.
7. L'internaute peut visionner la page et le résultat de sa requête sur son navigateur internet.

2.1.2 Différence entre CGI et module d'Apache

Comme nous l'avons vu au début de ce chapitre, il y a deux solutions pour interpréter un script sur un serveur web :

2.1.2.1 Par un module d'Apache

C'est la technique qui sera employée pour interpréter le code php de PMB. Le serveur va interpréter le script directement si il est configuré pour utiliser son propre module interpréteur (mod-perl pour le perl, mod-php4.so pour le php). Ce module est injecté dans le logiciel du serveur Web (Apache) qui effectue directement le travail d'interprétation. Il ne fait donc pas appel à un interpréteur séparé.

2.1.2.1.1 Avantage : Plus économe en mémoire et en ressources.

2.1.2.1.2 Inconvénient : On est obligé d'utiliser la version du module interpréteur chargé dans le serveur Web.

Pour pouvoir utiliser une autre version, on aura généralement recours au CGI ou à d'autres techniques plus complexes.

2.1.2.2 En tant que script CGI

Le serveur fait appel à un interpréteur installé sur le système de fichiers (/usr/bin/perl pour le perl /usr/bin/php4 pour le php) comme si il était appelé en ligne de commande. C'est ce qu'on appelle le CGI (Common Gateway Interface). C'est la technique qui est employée pour faire fonctionner Koha.

2.1.2.2.1 Inconvénient : Pour chaque script exécuté, (ce qui correspond approximativement à la connexion d'un client au serveur Web³) un interpréteur est chargé en mémoire. Cette méthode est donc plus gourmande en ressources.

³C'est une vision simplifiée, pour rendre la compréhension plus facile, j'ometts volontairement de parler d'exécutions parallèles appelées thread. Ce détail n'est pas nécessaire à la compréhension du principe.

2.1.2.3 Conséquences

Il faut toutefois relativiser, vu la puissance des machines actuelles, ces détails d'implémentation, bien que fondés sur des principes primordiaux pour les développeurs, ont une importance secondaire pour le choix d'un logiciel SIGB au niveau utilisateur.

En définitive, vu la proportion insignifiante du budget matériel dans un tel projet, c'est un critère qui ne sera pas pris en compte.

La différence théorique de consommation en ressource entre les deux procédés n'est pas suffisante que pour en faire un critère de sélection dans le cadre d'une application SIGB.

2.1.3 Stabilité d'une application Web du côté du client

Koha et PMB étant des applications s'exécutant du côté serveur, les causes de plantage ne peuvent être que :

- L'exécution du javascript. Javascript est un code qui s'exécute sur le navigateur du client afin de rendre les pages plus dynamiques. Il permet par exemple de vérifier le contenu d'un formulaire ou d'ouvrir une fenêtre pop-up ;
- Un plantage du navigateur.

2.1.4 Stabilité d'une application Web du côté du serveur

Imaginons un plantage au point 6 du paragraphe 2.1.1 dû, par exemple, à un bug dans le script.

Dans le cas où le script s'exécute en tant que module d'Apache : C'est l'exécution du script qui échoue mais la stabilité du serveur/interpréteur (Apache dans ce cas) n'est pas affectée, un fichier html contenant un message d'erreur est parfois envoyé au client, un log d'erreur est alors écrit sur le serveur, et Apache se met en attente d'une nouvelle requête.

Dans le cas où le script s'exécute en tant que CGI : Un processus interpréteur tel que `/usr/bin/perl` exécute le script comme un simple programme en local. En cas de plantage, ce processus se termine prématurément. Les ressources mémoire inhérentes au chargement de l'interpréteur sont libérées, un message d'erreur est envoyé au client, un log d'erreur est écrit, et le serveur se met en attente d'une nouvelle requête.

Une erreur d'exécution de script d'un service web (CGI ou module d'Apache) bien que décevante pour l'utilisateur, est moins compromettante que dans le cas d'un binaire monolithique s'exécutant chez le client. Dans ce cas c'est la stabilité de toute l'application qui serait compromise.

En conclusion : dans les deux cas, la stabilité du système n'est pas compromise par le plantage d'un script. Et il n'y a pas de risque de perte de données, autre que celles encodées dans la page en cours.

Pour rappel, Koha et PMB utilisent des interfaces Web, aussi bien pour l'interface bibliothécaire que pour l'OPAC. Lorsque l'on encode des données dans l'interface bibliothécaire, il s'agit d'un formulaire d'une page Web.

Pour que l'information soit prise en compte, il faut envoyer la page au serveur. Le script du SIGB va alors récupérer les informations contenues dans le formulaire de la page Web pour modifier la base de données. Un plantage du script ne causera donc que la perte des données contenues dans cette page.

2.2 Koha sous Linux

2.2.1 Stabilité de Koha

Au cours des tests, j'ai eu l'occasion de constater quelques bugs et plantages de scripts, je les ai signalés soit sur la liste de diffusion⁴ soit sur la page de rapport de bug de Koha⁵.

Je n'ai constaté aucune défaillance qui remette fondamentalement en question ce logiciel. Il ne serait pas utile de les énumérer ici puisqu'il suffit de faire un rapport de bug, et celui-ci est rapidement corrigé par l'équipe de développement.

Pour information, j'ai reçu les avis de correction de mes rapports de bug entre une et deux semaines après leur émission. Le dynamisme de l'équipe de développement est donc remarquable.

Ce point fut déterminant dans le choix du logiciel.

2.2.2 Stabilité de Koha du côté du navigateur (client)

Les développeurs de Koha recommandent d'utiliser le navigateur internet Firefox⁶ du moins en ce qui concerne l'interface bibliothécaire.

Toutefois de nombreux problèmes à propos de "bug" ont été rapportés par les utilisateurs d'Internet Explorer lors de l'affichage de l'interface d'administration.

Un site Web est par définition publique et doit pouvoir être visité à l'aide de n'importe quel navigateur.

Il faut bien comprendre que l'interface bibliothécaire de Koha n'est pas un site Web, mais est une application à part entière utilisant une interface Web.

Dans le jargon client/serveur on appelle cela un client léger. Dans ce cas, l'interface bibliothécaire de Koha a été développée pour un navigateur particulier, Firefox en l'occurrence.

Les codes devant s'exécuter en local utilisent une API^a dont l'implantation est spécifique au navigateur.

Cette API sera par exemple Le DOM (Document Object Model) du langage Javascript, partiellement ou mal implémentée par le navigateur Internet Explorer, alors qu'elle fait partie intégrante du langage Javascript depuis plusieurs années.

^aUne API est ce qui est utilisé par un programmeur pour développer une application. C'est l'abréviation de : application programming interface.

Il faut donc bien distinguer :

- **la navigation sur l'OPAC, qui elle est publique et fonctionne avec tous les navigateurs.**
Je l'ai testée avec un navigateur en mode texte (lynx, w3m, etc). L'OPAC fonctionne normalement. Il est donc utilisable pour les personnes malvoyantes utilisant des terminaux à

⁴infos@koha-fr.org et koha@lists.katipo.co.nz

⁵<http://bugs.koha.org/>

⁶<http://www.mozilla.org/>

synthèse vocale.

- De l'**interface bibliothécaire, qui a été développée spécifiquement pour Mozilla Firefox. Ce navigateur existe pour tous les systèmes d'exploitation et est totalement libre et gratuit.**

Dans un environnement standard (un pc équipé du navigateur Firefox), **je n'ai constaté aucun plantage du côté client.**

2.2.3 Performances et matériel requis

Des tests ont été effectués sur un Celeron 400 avec 64 Mo de RAM muni d'une carte réseau 10Mb. Même avec la charge très réduite d'un seul client simultanément, j'ai constaté quelques légères lenteurs par manque de mémoire RAM et aussi par un ralentissement du transfert dans le réseau local dû à la lenteur de la carte réseau.

Comme je l'ai expliqué au paragraphe 2.1.1, je rappelle que si 20 personnes lisent en même temps le résultat de leur recherche affiché sur l'écran, ceci ne constitue pas 20 connexions simultanées. A ce moment précis, il se pourrait même que le serveur soit en attente d'une connexion.

En effet une connexion dure le temps de :

- la réception de la requête sur le serveur,
- l'exécution du script,
- l'envoi de la page de résultat de la requête.

A la seconde où le serveur a envoyé la page au navigateur (le client), la connexion se termine. Pour qu'il y ait 2 connexions simultanées, il faut qu'il y ait deux requêtes en cours d'exécution.

L'internaute peut s'accorder tout le temps qu'il le souhaite pour lire la page Web résultant d'une requête, il n'est pas connecté et ne consomme aucune ressource sur le serveur durant ce temps.

Des tests de Koha ont été effectués sur un Celeron 800 équipé de 256 Mo de RAM, cette configuration est suffisante dans ces conditions, c'est-à-dire un seul utilisateur simultanément, peut-être deux ou trois à tout hasard si plusieurs utilisateurs valident au même instant.

Pour s'assurer de la réactivité du système, je recommanderai un P4 simple core ou bien hyperthread⁷. L'hyperthread est un bon choix aussi pour autant que l'informaticien qui installe le système, réinstalle un noyau Linux capable d'en exploiter toutes les performances.

Il en va de même pour les architectures à base d'un Athlon 64⁸ offrant un bon rapport qualité prix. De plus les instructions PowerNow⁹ de l'AMD constituent un avantage non négligeable dans les pays chauds. Cette technologie permet de réduire la consommation du processeur et donc son échauffement quand il n'est pas sollicité.

Du point de vue du rapport dissipation thermique/puissance de calcul, j'octroie une mention toute particulière à L'AMD Opteron 140 EE. Sa consommation de 30 Watt en fait le processeur idéal pour fonctionner dans un pays chaud ne disposant pas de salle climatisée.

⁷L'hyperthreading est une technologie qui permet d'obtenir la puissance d'une architecture à deux processeurs en mettant deux coeurs sur le même processeur

⁸http://www.amd.com/fr-fr/Processors/ProductInformation/0,,30_118_8796,00.html

⁹http://www.amd.com/fr-fr/0,,3715_12353,00.html

<http://www.amdcompare.com/us-en/opteron/details.aspx?opn=OSB140CSP5AT>

Quant à l'Opteron classique sans option EE (30 Watt) il consomme entre 82,1 et 84,7 Watt. Consommation électrique et dissipation thermique étant liées, à titre de comparaison, la dissipation thermique du Pentium 4 varie entre 55.3 et 115 Watt suivant sa cadence de respectivement 1.40 GHz à 3.80 GHz.

<http://support.intel.com/support/processors/pentium4/sb/CS-007999.htm>

Quant à la consommation du Xeon d'Intel cadencé à 2.80 GHz, sa consommation est de 111 Watt.

Voir paragraphe 6.1.1 Thermal Specifications page 68 de ce document :

<ftp://download.intel.com/design/Xeon/datashts/30235501.pdf>

Il faut aussi savoir qu'un AMD Opteron à cadence égale a une puissance de calcul nettement supérieure au Pentium 4. La cadence n'est donc pas directement proportionnelle à la puissance de calcul.

512 Mo de Ram seront amplement suffisants, une unité de backup et un UPS. Des disques durs hot swap en Raid ne seraient pas superflus car ils permettent de palier à la défaillance de l'un d'eux en le retirant avec un tiroir et en le remplaçant par un autre sans arrêter le système.

Koha étant une application Web, la puissance requise est proportionnelle au nombre de connexions simultanées.

Nous venons de voir qu'une personne lisant le résultat d'une recherche ne constitue pas une connexion, et ne génère aucun travail du côté du serveur. Donc le besoin en puissance n'est pas énorme.

Par contre l'échauffement du processeur me paraît être un critère important pour les pays du sud si le serveur ne se trouve pas dans une salle climatisée.

AMD et Intel vantent les mérites de leurs processeurs en se basant uniquement sur la puissance. Pour le Pentium 4 et le Xeon de chez Intel, les Athlon XP et 64 de chez AMD, la consommation électrique, donc la dissipation thermique, n'est pas prise en compte.

Concernant ce critère, ils affichent tous deux des spécifications allant de extrêmement médiocres, pour les Athlon¹⁰ et les Pentium 4 et Xeon à médiocre pour l'Opteron 140 EE dont la consommation de 30 Watt reste excessive.

Ce sont des processeurs présentant un risque d'échauffement. Ils exigent des dissipateurs thermiques (radiateurs) très performants pour pouvoir fonctionner sans danger.

Il existe des processeurs optimisés pour minimiser la consommation électrique (chauffant moins) mais aux performances inférieures :

– Le Géode de chez AMD

http://www.amd.com/us-en/ConnectivitySolutions/ProductInformation/0,,50_2330_9863,00.html

– L'Efficeon de chez Transmeta

<http://www.transmeta.com/efficeon/>

– Le C7 de chez VIA

<http://www.via.com.tw/en/products/processors/c7/>

Les deux derniers (C7 et Efficeon) arrivent presque à une consommation de 20 Watt en pleine charge, même si leur consommation devient extrêmement faible (entre 1 et 3 Watt) avec une faible activité, notre souci principal étant d'éviter un échauffement excessif du processeur,

¹⁰entre 67 et 89 Watt, cf <http://www.amdcompare.com/us-en/desktop/Default.aspx>

pas d'économiser du courant.

Il ne conviennent pas parfaitement pour être utilisés dans un serveur ou l'activité des ventilateurs et des disques durs est constante. De plus l'Opteron 140 EE en pleine charge ne consomme que 10 Watt de plus (30) et développe une puissance de calcul nettement supérieure.

En conclusion :

l'AMD Opteron 140 EE est, à mon avis, la moins mauvaise solution dans le domaine des processeurs pour serveur compatible x86.

Quant au P4 et Xeon de chez Intel, leur consommation étant très importante (plus de 100 W), ils risquent de surchauffer, tout comme les Athlon XP et 64 de chez AMD.

Dans tous les cas, il faudra prévoir des dissipateurs thermiques très performants même si leur prix dépasse les 60 euros, cet investissement est une bonne précaution pour éviter d'endommager le système par une surchauffe.

Il m'est impossible de recommander un dissipateur thermique, car je n'ai aucune compétence en la matière, toutefois, la firme UNIPRESS <http://www.unipress.be/> spécialisée dans les serveurs propose les produits de chez COOLER MASTER

http://ecd.coolermaster.com/products/products.php?_3ulist

Je recommande de monter le serveur dans une tour, plus apte à refroidir que les racks 1U. Les racks 1U ultra plats sont conçus pour être superposés et optimiser l'encombrement dans les salles de serveurs. **Ils sont à éviter car les composants y sont confinés.**



*Voici ce qu'est un rack 1 U.
A éviter !*

2.3 Le modèle de développement de Koha

2.3.1 Le design de l'interface de koha.

Le développement Koha est basé sur l'usage des templates ¹¹.

Pour créer un template, on demande généralement à un designer de créer des pages html¹² sans se soucier de la programmation.

Cette page html sera totalement statique, ce qui signifie que si on l'ouvre dans un navigateur internet, elle affichera toujours les mêmes éléments.

Cette technique permet à des personnes non spécialisées en informatique de dessiner l'interface de Koha avec des pages statiques et ensuite d'enregistrer leur travail dans des fichiers contenant des balises html complexes.

¹¹Un template est un fichier qui contient des balises html et des balises de template qui se chargent de l'affichage (voir paragraphe 2.3.1.1)

¹²Pour rappel, le Html est le langage de balises utilisé pour afficher des données sur Internet

2.3.1.1 Les balises de templates

Le développeur des scripts de Koha n'a alors plus qu'à remplacer, dans cette page Html, le contenu qu'il désire rendre dynamique par des **balises de templates**.

Ce contenu dynamique correspond généralement au résultat de l'interrogation d'une base de données dans un langage de programmation (Perl dans ce cas-ci). Ce résultat viendra s'afficher à la place des balises de template, sa mise en page étant assurée par les balises Html créées par le designer.

Après avoir ajouté des **balises de templates** dans la page html, il faut veiller à l'enregistrer avec l'extension appropriée (.tmpl). Ce fichier devient un template utilisable par les programmeurs de Koha pour générer l'affichage.

Cette technique permettra de créer un template pour chaque page de Koha.

Par exemple pour la circulation, on trouvera 4 templates :

- *branchtransfers.tmpl pour la page intitulée "Transfers"*
- *circulation.tmpl pour la page principale intitulée "Issues"*
- *returns.tmpl pour la page du retour intitulée "Returns"*
- *selectbranchprinter.tmpl pour la page de configuration de l'imprimante intitulée "Select Printer and Branch Settings"*

Pour d'avantage d'informations sur l'usage des templates dans Perl, n'hésitez pas à consulter la documentation suivante :

HTML : :Template - Module Perl pour utiliser les modèles HTML depuis des scripts CGI
<http://perl.enstimac.fr/ModulesFr/Html-Template.html>

Il existe deux bonnes méthodes pour séparer la représentation visuelle du langage de script :

- les technologies XML et XSLT, moins aisées à comprendre au départ, mais plus productives par la suite car elles augmentent la vitesse de développement.
- les templates, plus simples et efficaces au premier abord, ce qui permet à un programmeur de s'intégrer plus aisément dans l'équipe de développement. C'est un critère primordial (et plus important que la productivité, lorsqu'on est dans un projet non commercial) pour l'expansion du logiciel libre. Cette approche est donc la meilleure dans ce cas.

2.3.2 Le développement par modules.

Koha utilise des modules (paquetages)

Ces modules contiennent des fonctions prédéfinies destinées à résumer le codage et à diviser la difficulté de la programmation en un nombre suffisant de procédures pour la rendre aisément compréhensible. On placera dans ces modules les fonctions souvent utilisées dans la programmation de Koha pour les rendre réutilisables.

Ces modules sont de deux sortes :

- ceux fournis par le langage Perl ;
- ceux appartenant à Koha.

Davantage d'information sur l'usage des modules se trouve en annexe au paragraphe [E.3.2](#)

Pour davantage d'informations sur l'usage des modules en Perl consultez la documentation suivante :

perlmod - Modules Perl (paquetages et tables de symboles)

<http://perl.enstimac.fr/DocFr/perlmod.html>

Il va de soi que plus aucun programmeur n'entreprendrait un projet important, qui ne soit pas bien réparti en modules, et qui ne fasse pas appel à des bibliothèques existantes. Le principe est de ne pas réinventer la roue.

2.3.3 La répartition des composants

Koha est réparti en quatre parties :

- Le code de l'OPAC ;
- Le code de l'intranet ;
- Le code du démon Z3950 ;
- Le code qui accède à la base de données.

2.3.3.1 Les paquetages

Chaque composant de Koha est lui même structuré en paquetages.

Par exemple l'OPAC est divisé en deux paquetages :

- Le paquetage principal des scripts Perl de l'OPAC
- Celui des templates de chaque page HTML de l'OPAC

On a ainsi un code bien structuré en répertoires dans l'arborescence de Koha et dans lequel il est aisé de se déplacer.

Le fait de séparer les scripts d'accès à la base de données de ceux de l'affichage et de les utiliser dans un fichier principal (comme précisé dans le paragraphe [2.3.3](#)) fait penser au modèle de développement réutilisable appelé : Modèle Vue Contrôleur.

Cette forme de structuration moins aisée à comprendre au premier abord donne à long terme un code facile à maintenir, à étendre et aisément modifiable. Il est aussi plus lisible, car le code qui s'occupe de la logique n'est pas encombré de problèmes inhérents à l'affichage.

Par exemple, si l'on décide d'une restructuration complète des données ainsi que d'un changement de moyen de les stocker (Les développeurs pourraient vouloir passer à XML) les développeurs n'auraient que le composant DB (Le code qui accède à la base de données) à modifier. Le reste de l'application en est totalement indépendant.

Dans ce type de développement, les langages sont bien séparés. On ne verra jamais une balise Html dans du code Perl.

Ni jamais le moindre bout de code Perl dans un fichier de template contenant des balises Html(en théorie).

Cette méthode de répartition permet aisément de créer des templates pour chaque langue.

2.3.4 Suggestions sur le développement de Koha

Le contenu statique des templates

Il aurait probablement été plus judicieux de placer le contenu textuel statique dans une classe dédiée, qui serait instanciée par une fabrique abstraite [4] (Abstract factory¹³). Ainsi une phrase est remplacée par une variable de template qui reste identique quelle que soit la langue. Le traducteur n'a alors plus à se soucier des templates et des balises Html.

Il se retrouve devant un fichier ayant la forme :

Dans une classe française :

```
UneVariable="La phrase qui lui correspond en français" ;
```

Dans une classe anglaise :

```
UneVariable="The corresponding sentence in English"
```

Dans une classe allemande :

```
UneVariable="Der entsprechende Satz auf Deutsch"
```

Les avantages du contenu statique (phrases qui font partie de l'interface, libellé de champs, etc.) dans les templates sont qu'il est plus économe en ressources et plus simple à créer. Plus il y a du contenu statique moins il faut de balises de templates qui sont des variables remplacées par du contenu dynamique. Un de ses inconvénients est, par exemple, le fait qu'il est difficile à traduire.

Pour les phrases dans des variables insérées par des balises de templates, parmi les inconvénients, on peut citer : il faut charger un objet message dans la mémoire du serveur, un peu plus complexe à comprendre si on utilise une fabrique abstraite.

Cependant, elles permettent une plus grande séparation, donc plus grande facilité à maintenir et à traduire.

Remarque :

Si les développeurs de Koha voulaient encore augmenter la séparation entre la Vue (La page web dynamique dans ce cas) et la logique (du script Perl), ils pourraient passer par le langage XML avec des feuilles de styles XSLT. Grâce aux technologies XML, le code de l'affichage se résume parfois à une seule ligne.

Le rôle d'afficher le contenu par des template est alors déplacé dans la feuille de style XSLT. Une description détaillée du développement avec XML dépasse le cadre de cette étude.

2.3.5 Conclusion :

En conclusion, on peut dire que le modèle de développement de Koha est simple, robuste et efficace.

¹³Une fabrique abstraite est un modèle de conception réutilisable, servant à concevoir efficacement une application. Voir [4]

2.4 Efficacité des fonctions

Je vous présente ici une critique de certaines fonctions de Koha, j'ai fait ces observations en tant qu'informaticien sans compétences particulières en bibliothéconomie.

2.4.1 Catalogage :

1. Si on désactive la gestion du MARC (Paramètres => préférences système => marc), on a une interface d'encodage plus conviviale. Mais seul le titre est obligatoire sur la première page. Après avoir envoyé la page, on crée un enregistrement provisoirement incomplet et inutilisable dans les tables "biblio" et "marc_subfield_table".

Si on n'a pas pu terminer l'encodage suite à une erreur, (Si, par exemple, on a utilisé un code à barre déjà existant et qu'on quitte cette page) il est impossible de retrouver cette notice incomplète tant qu'une entrée n'a pas été faite dans la table "biblioitems".

Résultat :

Chaque fois que l'on se trompe dans l'encodage on crée un enregistrement inutilisable qu'on ne sait plus retrouver pour le compléter. Mais une requête SQL montre bien la présence de ces données.

Mais ce n'est pas vraiment gênant pour la raison suivante :

Dans ce formulaire on encodera tout au plus 20 mots. Si on prend une moyenne sur-estimée de 10 lettres par mots, on se retrouve avec un encombrement de 200 Octets à chaque fois que se répète le problème. Il faudrait que ça se répète 5 fois pour occuper 1 Ko 5000 fois pour occuper 1 Mo. Si on admet qu'à partir d'un Go le problème devient gênant, il le deviendra après 5 millions de fois, ce qui n'arrivera pratiquement jamais. Et même dans ce cas, une simple requête SQL, éliminerait aisément ces encodages perdus.

2. Un catalogueur compétent sait peut-être si il faut ou non mettre les tirets dans l'ISBN ? Il aurait fallu un petit javascript sur la page pour vérifier qu'ils n'y sont pas avant de pouvoir envoyer la page. Et à plus forte raison du côté serveur pour éviter de créer une entrée dans la base de données. Aucune vérification n'a été prévue. On aura donc des enregistrements avec et sans tiret selon la fantaisie et la compétence du catalogueur. Plus grave encore, aucune vérification n'est effectuée sur l'encodage de l'ISBN. Un ISBN ne peut contenir que des chiffres, un tiret suivi d'un chiffre de contrôle représenté par X si il vaut 10. Il existe des algorithmes permettant de vérifier la validité d'un ISBN, son codage en Javascript du côté client serait une chose aisée pour l'interface bibliothécaire. A plus forte raison en Perl sur le serveur, rien de cela n'est prévu ou ne fonctionne, j'ai pu encoder n'importe quoi dans le champ réservé à l'ISBN. Pour de plus amples renseignements sur les algorithmes de contrôle de l'ISBN : <http://www.isbn-international.org> Il existe un manuel traduit en français disponible à cette adresse : www.isbn-international.org/fr/download/ISBNmanualfr.pdf
3. Si par la suite on désire compléter une notice, on est obligé de le faire dans un formulaire au format MARC, il n'est plus possible de l'éditer via un formulaire simplifié.

4. Pour le catalogage, l'aide contextuelle est limitée à la recherche mais inexistante pour l'encodage des notices.

L'analyse technique de la fonction de catalogage m'a fait remarquer un certain nombre de problèmes. Toutefois, les versions de Koha changent plusieurs fois par trimestre et des améliorations sont apportées très rapidement à la version précédente en plus des corrections de bugs.

2.4.2 Circulation :

On peut supprimer la notice d'un livre en prêt, dans ce cas le système enlève le livre des emprunts du lecteur.

En cas d'erreur de manipulation de ce type, on ne sait plus dire qui a emprunté le livre.

2.4.3 La gestion des utilisateurs

L'interface est parfaitement claire et même sans être bibliothécaire, il est aisé de gérer les utilisateurs.

Toutefois, il n'est possible d'y encoder qu'une seule adresse. Koha étant destiné, dans le cadre de ce projet, à la gestion des bibliothèques universitaires, c'est un facteur limitant, sachant que la plupart des étudiants ont le domicile de leurs parents et celui de leur studio. La possibilité d'ajouter une photo et l'URL d'un éventuel site Web serait aussi une fonctionnalité intéressante.

2.4.4 Le client Z3950

2.4.4.1 Le script d'initialisation du démon Z3950

Sa qualité laisse à désirer pour les raisons suivantes :

- Il n'y a aucune vérification de l'unicité de l'instance du démon (pas de singleton).
Ce qui signifie qu'à chaque fois qu'on relance ce script un autre démon est lancé, ils peuvent donc s'accumuler. Mais cela ne crée pas de dysfonctionnement, c'est ce qu'on appelle des processus zombie, les systèmes de type UNIX tels que GNU/Linux peuvent le supporter sans planter.
- Rien n'est prévu pour stopper le démon. On est obligé de le faire avec une commande système d'arrêt brutal.

Pour ce faire il faut lister les processus en cours d'exécution avec la commande ps :

```
sh# ps -A
PID TTY          TIME CMD
11307 ?            00:00:00 postmaster
11308 ?            00:00:00 postmaster
11309 ?            00:00:00 postmaster
 5346 ?            00:00:00 dbus-daemon-1
 6661 ?            00:00:19 processz3950que
 9175 tty4        00:00:00 getty
10521 tty5        00:00:00 getty
```

```
9271 tty6      00:00:00 getty
5952 ?        00:00:00 fetchmail
```

Voir le manuel

- Le démon veut écrire sur le système de fichiers¹⁴ sans avoir de droits sur celui-ci...
Pour le faire fonctionner correctement, on est obligé d'accorder à son utilisateur¹⁵ le droit d'écrire dans le répertoire des logs.

2.4.4.1.1 En conclusion : une partie du travail est faite (le changement d'utilisateur et l'assignation de la configuration) mais le script d'initialisation demande à être perfectionné.

2.4.4.1.2 Solution : Il faut réécrire ce script d'initialisation en veillant à ce qu'il ne lance qu'un seul démon et que celui-ci puisse être stoppé.

2.4.4.2 Utilisation du client z3950

Si on a configuré Koha pour utiliser le MARC21, on est obligé de s'adresser à un réservoir en MARC21 tel que celui de la Library Of Congress. Si l'on désire utiliser un réservoir orienté vers la littérature francophone comme celui de la Bibliothèque Nationale de France qui utilise l'UNIMARC, il est indispensable d'avoir configuré Koha en UNIMARC lors de l'installation.

2.5 Structure des données

Les données sont stockées dans une base de données MySQL, aussi bien pour Koha que pour PMB. Bien sûr, des approches fondamentalement différentes peuvent être employées pour structurer celles-ci.

2.5.1 La structure des données de Koha

Les données usuelles telles que les emprunteurs sont stockées dans des tables qui leur sont propres avec des champs définis pour chaque information.

Les données relatives au catalogage sont gérées par une double structure MARC et non MARC.

On peut répartir la base de données de Koha par fonctionnalités.

Le catalogage dont la gestion est subdivisée en deux parties :

- le catalogage non MARC ;
- le catalogage au format MARC.

Cette partie de la base de données utilise le principe des **méta données**, qui est expliqué en annexe à la page [104](#).

Son application dans la base de données de Koha est décrite au paragraphe [E.1.6.3](#).

¹⁴Le démon écrit des logs qui sont des lignes dans un fichier texte décrivant les opérations qu'il accomplit et les problèmes qu'il rencontre

¹⁵www-data est le nom d'un utilisateur qui va sur internet pour se connecter à un serveur. A partir du moment où il est en contact avec le monde extérieur, cet utilisateur est potentiellement corrompible par un pirate. Partant de cette hypothèse, on ne lui accorde qu'un strict minimum de droits sur le système de fichiers

L'acquisition.**La circulation.****2.5.1.1 Le catalogage non MARC**

La structure d'enregistrement des notices non MARC utilise des champs prédéfinis dans des tables.

(Les notions de base concernant le modèle relationnel sont expliquées au paragraphe [E.1.5](#) en annexe).

Ces champs stockent les informations de base sur les ouvrages.

Il y a 7 tables qui sont dédiées au catalogage non MARC, elles contiennent le numéro du livre (Qui est la clé primaire¹⁶ de la table BIBLIO) et respectivement :

BIBLIO contient le titre et le nom de l'auteur et un numéro de livre qui sert en même temps de clé primaire pour cette table ;

ADDITIONALAUTHORS des noms de coauteurs ;

BIBLIOSUBJECT des entrées de sujet utilisées pour la recherche par mots clés ;

BIBLIOSUBTITLE un sous-titre ;

BIBLIOITEMS l'isbn, l'issn, la classification décimale de Dewey, le nombre de pages, la dimension ;

ITEMS le code à barre, le prix, l'abréviation de la bibliothèque principale ou sous bibliothèque ;

BRANCHES répète l'abréviation de la bibliothèque ou sous bibliothèque, son nom complet, son adresse, tél, fax, courriel.

La table BIBLIO est liée à la table MARC_SUBFIELD_TABLE par l'intermédiaire de la table MARC_BIBLIO¹⁷ qui a comme champs la clé primaire (Voir définition à la page [107](#) en annexe) de la table BIBLIO, les dates de création et de modification, etc...

2.5.1.2 Le catalogage au format MARC

La table MARC_SUBFIELD_TABLE¹⁸ reprend l'ensemble des libellés des champs et sous champs MARC et leurs valeurs sous forme de métadonnées (Pour plus d'informations concernant la notion de métadonnées consultez le paragraphe [E.1.6](#) en annexe).

On trouve aussi les tables :

- MARC_TAG_STRUCTURE et MARC_SUBFIELD_STRUCTURE qui sont la référence de la structure MARC choisie. Dans le cadre de cette analyse, il s'agit du MARC21.
- STOPWORDS qui contient les mots vides
- MARC_WORD qui sont les mots-clé de recherche
- AUTHORISED_VALUES qui contient les valeurs autorisées pour certains champs.

¹⁶Voir définition au paragraphe [E.1.7](#) en annexe

¹⁷Les personnes ayant des connaissances en bases de données relationnelles peuvent lire la note à la page [109](#) en annexe

¹⁸Sa structure est décrite au paragraphe [E.1.6.3](#)

Les métadonnées confèrent à la base de données une plus grande souplesse que si l'on avait assigné un champ dans la table pour chaque sous champ du marc. Cette structure sous forme "Nom du libellé/valeurs" peut s'adapter à tous les formats du MARC.

Il y a une répétition de certaines données entre les tables non MARC et les tables MARC, mais cela ne pose tout au plus qu'un problème de taille du disque dur et ne ralentit en rien l'application.

En effet, cette redondance des données ne viole aucune règle de normalisation¹⁹ puisque les données sont répétées dans des tables différentes.

2.5.1.3 L'acquisition

La redondance des données dans la table BIBLIO est utilisée pour servir de table de base pour le module d'acquisition.

Elle sert de référence aux tables :

- AQBOOKSELLERS qui sert de carnet d'adresses des fournisseurs par l'intermédiaire de la table AQORDERS.
- AQBUDGET qui comme son nom l'indique...

2.5.1.4 La circulation

La table ITEMS est liée, entre autres, à la table BORROWERS qui gère les lecteurs par l'intermédiaire de la table ISSUES qui gère les emprunts.

2.5.2 La structure des données de PMB

Il ne m'a pas été possible de faire une étude détaillée de la structure des données de PMB. Le logiciel lui-même possède son propre système de documentation, et permet de télécharger une image nommée "scheme.gif" de la structure de la base de données. Ces scripts de documentation doivent être appelés via un lien sous la forme :

<http://domaine.org/pmb/doc/noyau/>

Le modèle est composé de 60 tables.

Je ne dispose d'aucune documentation décrivant le développement de PMB²⁰. Une analyse d'un aussi grand nombre de tables sans disposer d'une bonne source d'information répartissant les tables par vues partielles serait un travail trop important.

Vous trouverez dans un document séparé une copie de la description de la structure des tables.

2.6 Koha, PMB et leurs SGBD

MySQL n'est pas le seul Système de Gestion de Base de Données relationnel libre. Il existe aussi PostgreSQL postgresql.org qui est souvent comparé à un équivalent libre d'Oracle. PostgreSQL est un système de gestion de base de données orienté objet d'une puissance et d'une stabilité extraordinaire. Si les développeurs d'application utilisent plus souvent MySQL que PostgreSQL, c'est pour les raisons suivantes :

¹⁹voir définitions en annexe au paragraphe [E.1.4](#)

²⁰comme l'excellente documentation sur l'architecture de Koha disponible sur kohadocs.org

- MySQL est plus léger et un peu plus rapide que PostgreSQL.
- MySQL est présent sur tous les serveurs LAMP (Linux, Apache, MySQL, Php) comme son nom l’indique. Les applications développées avec MySQL sont donc plus faciles à installer sur les serveurs mutualisés. La personne qui loue un espace sur ce type de serveur ne dispose pas de droit d’administration sur celui-ci, et ne pourra pas y installer de nouveaux logiciels. Le fait que MySQL est présent sur tous les serveurs LAMP influe donc fortement en faveur de son utilisation par les développeurs.
- L’orientation objet de PostgreSQL et l’ampleur de sa puissance seraient sous exploités, on choisit alors le SGBD le plus léger qui répond aux besoins de l’application.

Faute de temps, les possibilités de configurer Koha pour qu’ils utilisent un autre SGBD n’ont pu être étudiées. Ce n’est pas un point prioritaire puisque Koha a été développé pour utiliser MySQL et rien ne manque à ce SGBD pour assurer le bon fonctionnement de cette application.

En ce qui concerne PMB c’est impossible car les appels aux fonctions Php propres à MySQL sont effectuées 4577 fois dans la plupart des fichiers.

2.7 Le code de PMB

Ne disposant d’aucune documentation sur le sujet, je ne me serais pas permis de le juger de manière catégorique. Je n’ai parcouru que quelques fichiers de codes source.

A ma grande surprise, pour un projet d’une telle ampleur, aucune volonté de structurer le développement en modèles de conception réutilisables n’est apparente. Mais faute de documentation, je ne peux pas affirmer qu’ils ne sont pas utilisés.

Les classes (plus ou moins l’équivalent des modules de Koha) sont rangées dans le répertoire “pmb/classes”. Chaque action²¹ peut ainsi y faire appel.

La répartition du code en répertoires respecte celle des onglets “Circulation”, “Catalogue”, “Autorités”, etc. Il est donc très aisé de s’y retrouver.

Par exemple, le script `catalog.php` correspondant à l’onglet “Catalogue” inclut le script : `catalog/catalog.inc.php` situé dans le répertoire `catalog`.

Le script `catalog.inc.php` inclut lui même le script : `catalog/notices/notice_create.inc.php`

Chaque ligne de code est bien commentée, et en français.

La conversion des documents XML en iso2709 et vice versa se fait à l’aide d’une feuille de style XSLT.

L’avantage de cette méthode est de centraliser cette difficulté dans un seul fichier et de ne pas avoir à s’encombrer de code PHP pour pouvoir se concentrer sur le problème du format

²¹Les programmeurs doivent comprendre ce mot au sens VMC du terme [4]. Pour les autres lecteurs, les actions consistent à répartir le code en tâches et en structures logiques. Par exemple, le prêt d’un livre est une action qui déclenche d’un côté des appels de routines liées à la logique : mémoriser la sortie dans la base de données, vérifier que le nombre max d’emprunts n’est pas dépassé, etc, et d’un autre, des appels de routines liés à l’affichage à l’écran. Ce sens du mot “action” est donc assez proche du sens usuel

de l'ISO2709 .

Conclusion :

Le code est très simpliste, il n'utilise pas de modèle compliqué dont il est difficile de s'imprégner de la logique. Toute personne désirant s'investir dans le développement de PMB doit juste connaître PHP (pour le développement de la plupart des fonctionnalités).

La structure d'inclusion récursive en arborescence est claire et facile à comprendre.

Quitte à augmenter la complexité de la structuration du code, il serait judicieux de séparer (ou de le faire de manière plus voyante si c'est le cas) la partie logique du code de la partie visuelle (celle qui se charge de l'affichage à l'écran).

Chapitre 3

Sécurité

L'étude de la sécurité porte sur :

1. La sécurité de l'application OPAC et de l'interface d'administration :
 - (a) L'interface d'administration utilise-t-elle une connexion sécurisée en https ?
 - (b) Où et comment les mots de passe sont-ils stockés (en clairs ou cryptés) ?
2. La sécurité des données
 - (a) Les possibilités de sauvegardes propres à Mysql
 - (b) Les possibilités de sauvegardes fournies avec Koha
3. la sécurité du serveur http
4. la sécurité du serveur au sens général, c'est à dire :
 - (a) La mise en place d'un pare-feu.
 - (b) La sécurité du matériel (local, accès physique à la machine, panne matériel).
 - (c) La sécurité/stabilité du système d'exploitation.

3.1 Koha sous Linux

3.1.1 Sécurité de l'application

3.1.1.1 Stockage des mots de passe

Le login et le mot de passe de connexion à la base de données sont stockés en clair dans `/etc/koha.conf`. Ils sont donc situés en dehors du "DocumentRoot" (espace autorisé aux scripts de l'espace web). Autrement dit, un utilisateur mal intentionné ne peut pas voir/exécuter ce fichier en indiquant son chemin dans l'URL.

Bien sûr, le mot de passe stocké dans ce fichier ne circule pas sur le réseau, il est lu en local par les scripts perl qui en ont besoin pour se connecter au serveur SQL.

Il n'y a qu'un seul couple mot de passe/login de ce type. C'est un login correspondant à un utilisateur du serveur de bases de données.

Les autres mots de passe (des lecteurs/administrateurs de l'application) sont stockés dans la table "borrowers" (dans les champs "userid" et "password") de la base de données. Les mots de passe sont cryptés et stockés dans une chaîne de caractères (30).

La méthode de cryptage la plus courante est la comparaison de la somme md5 (algorithme de cryptographie) que l'utilisateur a tapé dans la zone d'édition de la page web et la somme md5 du mot de passe enregistré dans la base de données.

Cf. checkpw dans \$KOHA/intranet/modules/C4/Auth.pm

Cette méthode est sûre pour le stockage des mots de passe sur le disque du serveur mais n'offre aucune garantie pour leur transmission depuis le client. On est donc confrontés au fait que les informations (mots de passe compris) transitent en clair entre le client et le serveur. Ces informations pourraient être obtenues aisément par un "sniffeur", mais ce risque doit encore être vérifié.

3.1.1.2 Habilitations des utilisateurs

La différence entre un lecteur et un bibliothécaire se configure via les habilitations.

Lecteurs => Modif d'habilitation.

La sécurité à ce niveau se base sur l'interdiction d'accès à certaines fonctionnalités en fonction des habilitations.

Le problème qui se pose avec ce principe d'habilitation, c'est que si un utilisateur a l'habilitation : "borrowers Add or modify borrowers" lui permettant d'ajouter ou modifier les emprunteurs de la bibliothèque, il peut si il le désire, s'attribuer l'habilitation "superlibrarian Access to all librarian functions" qui lui donne un accès total au SIGB.

Il faut donc considérer dans ce cas que soit les deux habilitations font double emploi et sont une erreur de conception, soit on se base sur la confiance et les habilitations sont plus destinées à répartir les tâches des bibliothécaires et à éviter les erreurs de distraction.

De toute façon dans la globalité de sa conception, koha est un SIGB qui se base davantage sur la confiance que sur la sécurité. Par exemple une personne autorisée à gérer les préférences système peut aussi en supprimer et pas simplement les modifier. Il en va de même pour la configuration des champs du MARC, etc.

En conclusion : La gestion des habilitations des utilisateurs de Koha se base plus sur la confiance que l'on accorde aux bibliothécaires que sur un principe sécuritaire. Un bibliothécaire qui a le droit de modifier les emprunteurs peut s'auto proclamer administrateur avec tous les droits sur le SIGB. Il faudra veiller à attribuer ce droit uniquement aux bibliothécaires de confiance, méticuleux, et compétents.

Il en va de même pour les outils de configuration ou l'on peut aussi bien supprimer une option de configuration que la modifier. Le risque d'endommager le système par erreur (tag du Marc par exemple) est simplement limité par un avertissement. Même remarque il faudra veiller à attribuer ce droit uniquement aux bibliothécaires de confiance. C'est à cette condition que Koha pourra rester pleinement opérationnel.

3.1.2 Sauvegarde des données

Il existe de nombreuses possibilités d'effectuer une sauvegarde des données. Les premières possibilités décrites ci dessous sont propres au SGBD (Système de Gestion de Base de Données) Mysql.

Les autres sont fournies par des scripts de Koha.

L'énumération des options ainsi que les exemples ont pour but de mettre en évidence de manière objective le niveau de complexité des différentes possibilités de sauvegarde des données. Les aspects plus techniques seront décrits en annexe.

3.1.2.1 Les possibilités de sauvegardes propres à Mysql

3.1.2.1.1 Copie avec cp La plus simple consiste en une copie de tous les fichiers de tables (les fichiers *.frm, *.MYD, et *.MYI) au moment où le serveur ne met rien à jour (dans ce cas, il faut veiller à stopper le serveur le temps d'effectuer ces copies).

Des explications détaillées sont fournies en annexe au paragraphe [C.1.1](#)

Les avantages de cette technique sont :

- sa facilité d'exécution ;
- sa facilité de lancement par un gestionnaire de tâche.
Pour de plus amples renseignements sur la gestion différée des tâches sous GNU/Linux consultez le paragraphe [D.4](#) en annexe.
- sa facilité de fonctionnement. En effet, elle ne nécessite pas que le serveur de base de données, ni même le système d'exploitation fonctionne.
Par exemple, cette méthode peut donc être employée pour sauvegarder les bases de données d'un serveur en panne. Si c'est l'ensemble du système qui est défaillant, on peut enlever le disque dur de la machine et le monter sur une autre pour effectuer cette opération.

Les inconvénients en sont les suivants :

- La copie ne peut être restaurée que sur une version de serveur identique et risque d'être inutilisable après une mise à jour car la structure interne du système de fichier de la base de données peut changer d'une version à l'autre.
- Il y a obligation de copier l'ensemble de la base de données, la copie n'est donc pas sélective.
- Le volume des bases de données copiées est beaucoup plus important (plus de 4x) que celui d'un dump de ces mêmes bases.
Par exemple, pour une base de données Koha occupant 7,2 Mo, un dump de cette base occupe 1.5 Mo avec l'option “--complete-insert” (voir paragraphe [C.1.3](#)).
Ce dump comprimé avec bzip2, n'occupe que 160Ko, alors qu'une archive tar comprimée de la base Koha occupe 536Ko

La copie de la base de données peut être utilisée pour effectuer des dépannages mais pas comme solution de sauvegarde quotidienne des données.

3.1.2.1.2 mysqlhotcopy Le script Perl mysqlhotcopy offre la même possibilité, mais en posant automatiquement des verrous au moment de la copie des fichiers :
Pour davantage de détails sur l'utilisation de mysqlhotcopy consultez le paragraphe [C.1.2](#) en annexe.

Les avantages de cette méthode sont :

- sa facilité d'exécution ;
- sa facilité de lancement par un gestionnaire de tâche.
Pour de plus amples renseignements sur la gestion différée des tâches sous GNU/Linux consultez le paragraphe [D.4](#) en annexe.

- sa fonction de pose automatique d'un verrou pour assurer la cohérence des données.

Les inconvénients sont :

- La copie ne peut être restaurée que sur une version de serveur identique.
- On est obligé de copier l'ensemble de la base de données, la copie n'est donc pas sélective.
- Le volume des bases de données copiées est beaucoup plus important (plus de 4x) que celui d'un dump de ces mêmes bases.
- Contrairement à la copie manuelle qui peut se faire en montant le disque sur une autre machine, ce script exige que le serveur de base de données soit opérationnel puisqu'il s'y connecte.

La copie de base, qu'elle soit effectuée par une copie manuelle ou avec le script `mysqldump`, n'est pas une solution idéale de sauvegarde quotidienne des données à cause du volume des fichiers générés.

3.1.2.1.3 mysqldump La méthode de sauvegarde des données la plus courante est d'utiliser la commande `mysqldump` qui permet d'exporter une base ou un groupe de bases vers un fichier.

Cette commande ne s'utilise pas dans une console SQL mais dans un shell UNIX.

Les avantages de cette commande sont :

- sa facilité de lancement par un gestionnaire de tâches.
- la possibilité d'exporter sélectivement une base, ou une table.
- C'est le moins rébarbatif des utilitaires propres à MySQL, bien qu'il requiert quelques connaissances de la structure des tables si l'on désire effectuer une sauvegarde sélective.
- Le volume des archives n'est pas très important, il peut en plus être comprimé et restauré directement sous cette forme.

Les inconvénients sont :

- Cet utilitaire n'est pas aussi sélectif que la commande SQL "SELECT" : il ne peut porter que sur des tables entières et non des groupes d'enregistrements. Pour des opérations de sauvegarde de sécurité, cela ne pose cependant pas de problème.
- Il requiert quelques connaissances de la structure des tables si l'on désire effectuer une sauvegarde sélective.

mysqldump est la solution de sauvegarde qui sera employée dans la plupart des cas.

- Elle offre un bon rapport : facilité d'utilisation/souplesse.
- Elle permet d'effectuer la sauvegarde dans une transaction pour un instantané de la base dans un état cohérent.

Voir détail des options et exemples d'utilisation au paragraphe [C.1.3](#) en annexe.

3.1.2.1.4 SELECT Il est aussi possible de faire des sauvegardes sélectives en envoyant le résultat d'une requête dans un fichier :

L'avantage principal de cette solution est :

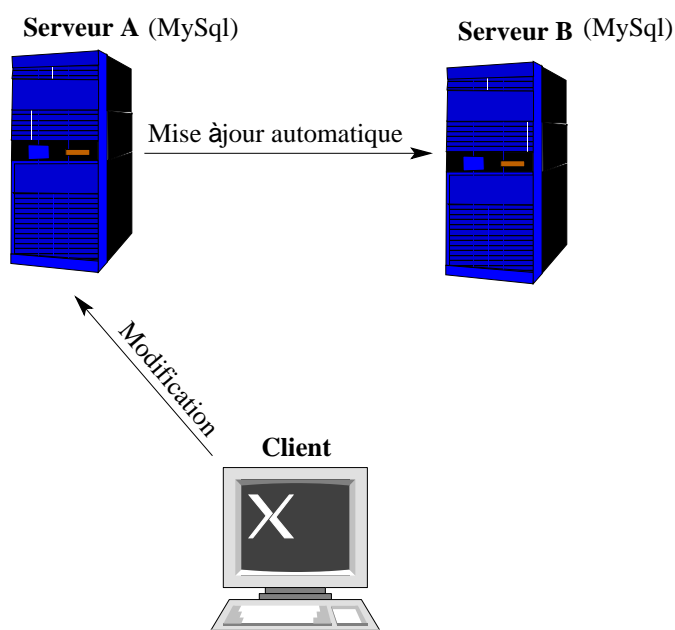
C'est une méthode très sélective. On peut, par exemple, l'employer pour récupérer certains comptes utilisateurs de Koha et les restaurer dans une nouvelle installation lorsqu'on effectue des tests.

L'inconvénient majeur est :

Elle requiert une intervention manuelle de connexion à la base de données ou une utilisation dans un script (Perl par exemple). Elle est donc plus spécifique et moins aisée à mettre en oeuvre via un gestionnaire de tâches.

Elle sera utilisée à bon escient lorsqu'on désire effectuer des sauvegardes spécifiques et occasionnelles. Cette méthode est décrite en détail en annexe au paragraphe [C.1.4](#)

3.1.2.1.5 La réplication des serveurs Répliquer **un serveur de base données**, c'est faire en sorte que les modifications des données qui y sont faites par le client sur le **serveur A** (maître) soient automatiquement répercutées sur le **serveur B** (esclave). Pour ce faire le maître écrit dans un fichier binaire toutes les modifications qui sont apportées à sa base de données. L'esclave n'a alors qu'à lire ce fichier binaire pour l'imiter.



La réplication permet :

- d’assurer une sauvegarde non bloquante (Cf. paragraphe C.1.4) des données sur l’esclave et non plus sur le serveur principal.
- d’augmenter la robustesse du système : si un problème survient sur le serveur maître, le serveur esclave peut être utilisé comme serveur de secours.
- d’augmenter la rapidité d’accès aux données par une répartition de la charge de traitement des requêtes entre le serveur maître et le(s) esclave(s).

Dans le cas où l’on désire simplement mettre en place une réplication pour assurer la sécurité des données, on n’effectuera pas de répartition des charges, donc on n’augmentera ni la rapidité d’accès aux données ni la robustesse du système.

Matériel requis

Pour assurer une simple fonction de sauvegarde : Un ordinateur de récupération, par exemple un PIII avec un bon disque dur fera l’affaire.

Si la réplication des charge est nécessaire : il est fort peu probable que la réplication des charges soit nécessaire pour un serveur hébergeant Koha pour une bibliothèque universitaire. Toutefois dans ce cas je recommanderai deux serveurs bi-Opteron équipé d’Opteron 240 EE¹.

Bien sûr, il faut disposer de plusieurs machines, la plus puissante servant de serveur maître sur lequel s’exécuteront les requêtes de l’OPAC et de l’interface d’administration. Une autre machine moins puissante servira d’esclave de réplication. L’esclave ne reçoit aucune requête, il se contente de se mettre à jour en lisant les logs binaires du maître.

Les avantages de cette méthode sont :

- Quoi qu’il arrive au serveur maître, une sauvegarde des données peut toujours s’effectuer sur le serveur esclave.
- du moment que les deux ordinateurs sont connectés via le réseau de l’institution, toute l’infrastructure de sauvegarde peut être isolée dans un autre bâtiment, ce qui augmente la sécurité en cas d’incendie, d’effraction, etc.
- Permet de faire des sauvegardes non bloquantes des données sur l’esclave et non plus sur le serveur principal. Ce dernier n’est pas perturbé.

Les inconvénients de cette solution sont :

- L’utilisation du log binaire destiné à la réplication ralentit le serveur maître d’environ 1%, ce qui est quasi négligeable.
- Il est nécessaire de disposer d’une deuxième machine (même de très faible puissance) pour l’esclave.
- C’est une solution de sauvegarde un peu plus compliquée à configurer.

Malgré ces difficultés de configuration, les avantages de la réplication en font une solution intéressante. Toutes les étapes de sa mise en oeuvre sont décrites au paragraphe C.1.5 en annexe.

3.1.2.1.6 Solutions proposées Les solutions propres à MySQL les plus adaptées à une sauvegarde régulière des données sont :

¹Pour rappel les Opteron de la gamme EE consomment 30 Watt au lieu de 82,1 (voir paragraphe 2.2.3) et ils ont la même puissance, comme ils consomment moins, ils chauffent moins et sont mieux adaptés pour fonctionner sans salle climatisée dans les pays chauds.

La sauvegarde avec mysqldump (voir paragraphe 3.1.2.1.3)

Pour mysqldump, le serveur est équipé soit :

- d'un disque dur externe

Cette solution n'est pas très adaptée si l'on désire effectuer des sauvegardes de plusieurs années, de plus elle ne met pas les données à l'abri d'une erreur de l'opérateur de sauvegarde (par exemple effectuer une suppression récursive à partir d'un répertoire parent alors qu'on se croyait dans un sous répertoire). Le même support concentre toutes les données en cas d'accident, défaillance du disque, perte, c'est l'ensemble des données qui sont perdues.

- d'un graveur (CD/DVD) Les supports peuvent (doivent) être dupliqués (voir politique de sauvegardes). Certains d'entre eux (les archives complètes) devront être stockés dans un autre bâtiment sous la responsabilité d'une autre personne. L'avantage de cette technique est le faible coût des supports, l'accès direct aux fichiers et la robustesse.

- d'un DAT de 2 à 24 Go (selon lecteur)

Ce support est coûteux et ne permet pas d'accès direct. De plus, l'usure est assez rapide, 2000h pour les têtes et 20000 passages pour les bandes.

- DDS : Digital Data Storage jusqu'à 40 Go.

- SDLT : Super Data Linear Tape de 110Go à 1,2To

Il est aussi envisageable que l'opérateur de sauvegarde ne dispose pas du support physique, et dépose par le réseau ses archives sur un serveur de fichiers, il délègue alors à l'administrateur du serveur de fichiers le soin d'effectuer les sauvegardes sur un autre support.

La réplication des serveurs (voir paragraphe 3.1.2.1.5)

Le matériel nécessaire pour la réplication des serveurs est le suivant :

1. Un ordinateur maître d'une puissance adaptée à la charge qu'impose la bibliothèque (voir paragraphe 2.2.3).
2. Un ordinateur esclave de réplication d'une puissance quelconque.
 - (a) La tâche de cet ordinateur se limitera à la lecture des logs binaires écrits par le maître, il ne doit répondre à aucune requête de sélection des données.
 - (b) L'ordinateur esclave doit être relié au maître soit par une connexion réseau permanente soit intermittente (ex PSTN + modem 56K).
 - (c) Celui-ci assurera la redondance et donc la sécurité des données. Les sauvegardes (voir plus haut pour les supports) se feront à partir de cet ordinateur.

Comme je l'ai indiqué au début du paragraphe 3.1.2.1.6 la réplication ne met pas les données à l'abri d'une erreur de manipulation, d'encodage ou de suppression, car dans ce cas l'erreur sera répliquée. Par contre si la réplication ne protège pas d'une suppression par erreur dans la base de données, elle protège d'une suppression par erreur sur le système de fichier de l'ordinateur et d'une perte de données due à un problème matériel.

L'objectif de la réplication est complémentaire à celui d'une sauvegarde avec mysqldump, qu'il est dès lors possible d'effectuer sur le serveur esclave sans perturber le serveur maître.

3.1.2.2 Facilité de mise à disposition des données devant être récupérées par un logiciel de sauvegarde.

Il existe un logiciel graphique (MySQL Administrator) qui facilite considérablement la maintenance de MySQL.

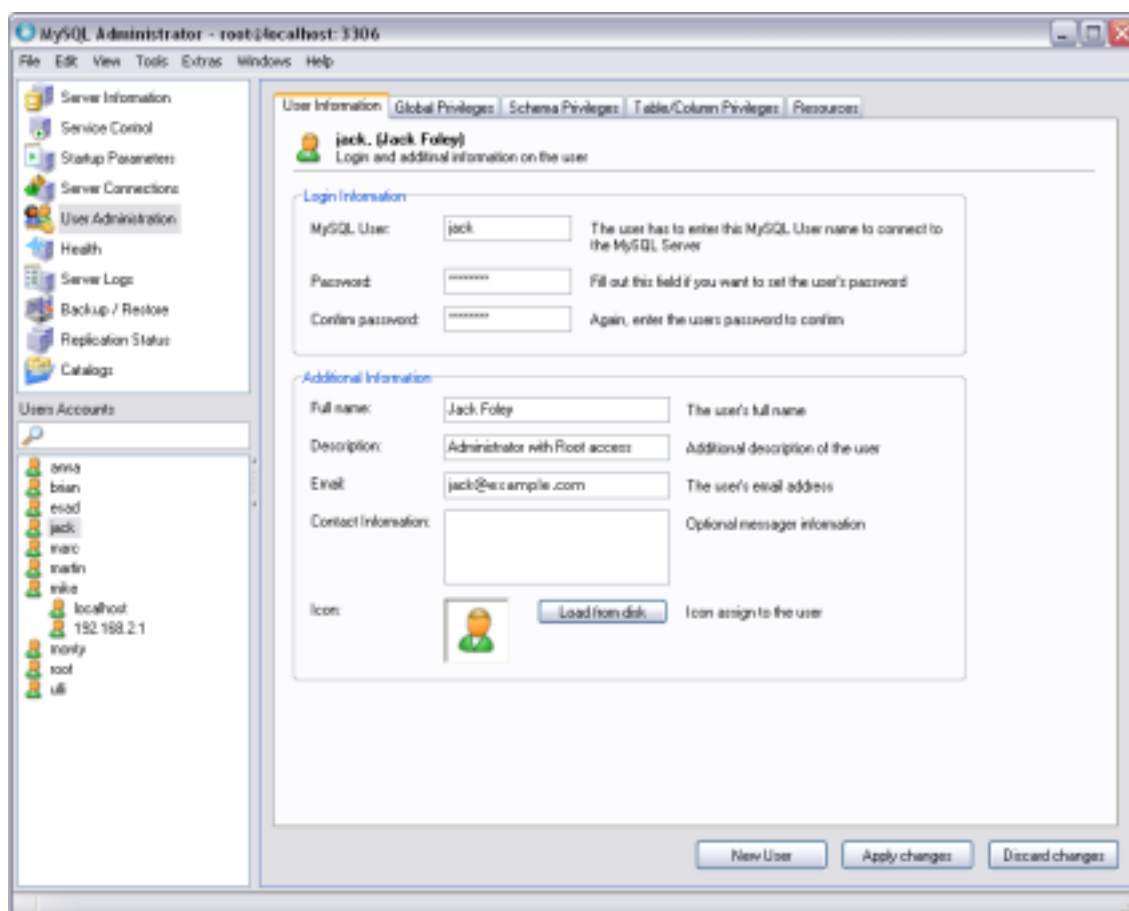
<http://www.mysql.com/products/tools/administrator/> Il ne faut pas le confondre avec `mysqladmin` qui l'outil d'administration par défaut de MySQL. <http://dev.mysql.com/doc/mysql/fr/mysqladmin.html> Ce logiciel n'a été testé que très sommairement : une sauvegarde des données s'est effectuée sans problème et le logiciel s'est connecté sans difficulté au serveur à partir de la machine de bureau sous GNU-Linux.

Ce logiciel existe pour les systèmes : Mac OS X, GNU-Linux et Windows.

Il a comme avantage de pouvoir être lancé en ligne de commande avec les mêmes options que les autres outils d'administration. Si l'on indique le nom de l'utilisateur, le mot de passe et le nom d'hôte de la machine dans la ligne de commande, la boîte de dialogue de connexion est évitée.

Ex :

```
sh$ mysql-admin -u nombase -p motdepasse -h nommachine
```



3.1.2.2.1 Effectuer une sauvegarde automatisée

Il faut choisir Backup dans le menu de gauche et donner un nom au projet.

Ensuite, il faut choisir les bases que l'on désire inclure dans l'archive. Il est possible, d'après l'interface, d'automatiser cette tâche : une ligne est alors ajoutée dans la crontab (voir paragraphe D.4.2) de l'utilisateur mais, malheureusement, la commande indiquée ne peut pas

fonctionner puisque les paramètres de connexion à la base de données ne sont pas indiqués.

Toutefois, si l'on n'utilise pas cette option avancée, l'archivage se déroule sans problème via une connexion distante. Le logiciel dépose l'archive dans le répertoire choisi par l'utilisateur².

Il y a d'autres options avancées similaires à celles décrites avec `mysqldump` (voir en annexe au paragraphe C.1.3), mais elles n'ont pas été testées.

3.1.2.2 Bref aperçu des autres possibilités Les autres fonctionnalités de ce logiciel sont :

- la gestion des répliqués ;
- la création et la modification des tables et des bases ;
- la gestion des utilisateurs, de la charge, etc.

En testant uniquement l'archivage des bases, l'installateur sera surpris par la puissance et la simplicité de ce logiciel qui n'est probablement qu'un frontal utilisant en arrière-plan les outils d'administration habituels. L'administrateur expérimenté et habitué aux outils en mode texte y trouvera la plupart des possibilités offertes par ceux-ci. En effet, le libellé des options porte un nom similaire aux options de la ligne de commande, ce qui n'est pas toujours le cas dans les frontaux destinés au grand public.

3.1.2.3 Les possibilités de sauvegardes fournies avec Koha

Les possibilités de sauvegardes sont de deux types :

- Les possibilités d'importation/exportation directement disponibles dans l'interface (Menu paramètres) sont décrites aux paragraphes 4.1 et 4.2.
- Celles fournies sous forme de script utilisable en ligne de commande.

Par convention, dans la suite de ce document, l'interface bibliothécaire de Koha est dans `/usr/local/koha/intranet`.

Les chemins des utilitaires de sauvegarde fournis avec Koha seront désignés de manière relative par rapport au répertoire d'installation.

Le chemin relatif "`scripts/misc/backup.sh`" désigne en réalité :

`/usr/local/koha/intranet/scripts/misc/backup.sh`

3.1.2.3.1 Le script `backup.sh`

Il se trouve dans `scripts/misc/backup.sh`. Pour l'utiliser, il faut l'éditer³ et modifier certains paramètres, ce qui lui enlève une grande part de son intérêt car il n'est pas directement utilisable, et sa modification exige des compétences informatiques.

- A la ligne 8 :

```
mysqldump -u koha -ppassword koha > $KOHA_DUMP &&
```

Il faut remplacer `password` par votre mot de passe.

- A la ligne 14, le script déplace l'archive dans un répertoire qui n'a pas été créé lors de l'installation. Ce script ne vérifie donc pas l'existence d'un répertoire avant d'y envoyer un élément, il faudrait le réécrire pour qu'il vérifie l'existence de ce répertoire et le crée si nécessaire.

²Dans un répertoire de la machine sur lequel tourne le logiciel bien sûr, pas dans le répertoire du serveur de base de données.

³L'utilisation de l'éditeur vim est décrite en annexe au paragraphe D.5

3.1.2.4 Politique de sauvegarde

Une politique de sauvegarde différentielle⁴ ou incrémentale⁵ est difficile à mettre en oeuvre sur une base de données, il faut passer par un dump complet de la base et ensuite enregistrer les différences observées par rapport au dump de la précédente sauvegarde. Il est peu probable que la base de données n'ait subi aucune modification d'une sauvegarde à l'autre. La solution la plus simple et la plus sûre est d'effectuer des sauvegardes complètes à intervalles réguliers. Tout dépend du rythme de modification des données. On peut ajouter au script de sauvegarde une ligne de code qui vérifie si le fichier d'archive est bien différent du précédent. A la différence d'une vraie sauvegarde différentielle, l'archivage est complet et se fait sur un seul fichier, ce qui signifie que tous les (ou aucun des) éléments (ne) sont sauvegardés. Dans ce cas, c'est une sauvegarde complète assortie d'une condition.

La solution suivante permet de faire face à la plupart de situations :

- Une sauvegarde complète toutes les nuits (1 jeu de supports, les CD ou DVD devront être réinscriptibles) avec écrasement cyclique, hebdomadaire (ou mensuel), les sauvegardes du cycle courant écrasent celles du cycle précédent.
- Le support du vendredi est conservé 1 mois comme sauvegarde hebdomadaire.
- Le support du dernier vendredi du mois est conservé 1 an comme sauvegarde mensuelle.
- Le support du dernier vendredi de l'année est conservé sans limitation dans la durée comme sauvegarde annuelle.

3.1.3 La sécurité du serveur http

Comme nous l'avons vu au paragraphe 2.1, les scripts de Koha s'exécutent sur un serveur Web. La sécurité de Koha dépend donc largement de celle du serveur http (Web) Apache en l'occurrence. Nous allons aborder la sécurité du serveur Apache.

Pour aller au-delà de ce qui sera présenté ici, une courte documentation est disponible sur le site d'Apache :

http://httpd.apache.org/docs/misc/security_tips.html#cgi

Certains aspects de la sécurité d'Apache sont configurés par le développeur du paquet Debian.

3.1.3.1 Identité sous laquelle est démarré le serveur

Apache est démarré par l'utilisateur root⁶ qui lance le processus père, mais tous les processus fils, qui répondront aux requêtes, appartiendront à l'utilisateur défini par la directive User. Une directive est un mot clé, suivi ou non de paramètres, permettant de configurer Apache en lui indiquant ce qu'il doit faire, comment il doit le faire et dans quelle condition il doit faire telle ou telle chose.

User www-data

Dans le fichier `/etc/apache/httpd.conf` cette directive est affectée à `www-data`.

⁴Seuls les fichiers modifiés ou nouveaux depuis la précédente sauvegarde sont sauvegardés, en remplaçant l'ancienne version.

⁵Idem que sauvegarde différentielle, sauf qu'on garde l'ancienne version des fichiers.

⁶root est un compte super utilisateur sur le système ayant tout les privilèges d'administration.

www-data est un utilisateur spécialement créé à cet effet avec des droits restreints. En cas d'intrusion, celle-ci ne pourra se faire qu'à partir de cet utilisateur. Et elle ne pourra s'étendre sur le système qu'avec ses permissions très restreintes. Ce qui veut dire que si un pirate effectue une intrusion sur le système à partir de cet utilisateur, il ne pourra presque rien faire car d'autres sécurités⁷ sont prévues pour protéger les scripts de Koha.

3.1.3.2 Répertoire racine à partir duquel httpd va distribuer les fichiers

La directive DocumentRoot est affectée à “/var/www” dans le fichier de configuration du paquet Debian. A défaut d'instruction, Apache distribue ses fichiers à partir du répertoire “/usr/local/apache/htdocs”.

En aucun cas cette directive ne peut pointer un répertoire supérieur à l'arborescence des fichiers de configuration. Il faut absolument veiller à éviter d'affecter à la directive “ DocumentRoot” la racine “/” du système de fichier car, dans ce cas, un utilisateur mal intentionné pourrait accéder aux fichiers de configuration.

3.1.3.3 Accès à partir de certains domaines uniquement

Il est possible de configurer Apache pour que certains répertoires situés sous l'arborescence de DocumentRoot ne soient accessibles qu'à partir de certains domaines.

Ainsi on pourrait décider que l'OPAC n'est accessible que de l'intérieur de la bibliothèque ou d'un intranet et l'interface d'administration pourrait être réservée à un autre réseau. Voir la directive <Directory> dans la documentation d'Apache :

<http://httpd.apache.org/docs/1.3/mod/core.html#directory>

Je recommande par mesure de sécurité, de limiter l'accès à l'interface bibliothécaire, à un certain sous domaine. Un exemple de sous domaine serait “biblio” dans un nom de machine complet tel que :

postetrav1.biblio.organisation.org

On peut par exemple, demander à Apache de limiter l'accès à l'interface bibliothécaire à 99 ordinateurs membres du sous domaine “biblio”, avec la directive <Directory>.

Pour rappel un sous domaine correspond à une classe d'adresse IP.

Une adresse IP est un numéro unique, identifiant un ordinateur dans un réseau. Pour des raisons informatiques (binaire) ce numéro est séparé par des points 139.165.158.59.

On peut créer des groupements (par sous domaines) dans l'attribution de ces numéros aux ordinateurs.

On peut ainsi déterminer que les postes de travail faisant partie de la bibliothèque vont de : 139.165.158.1 à 139.165.158.100, ce qui nous fait 99 numéros unique { 139.165.158.1 ; 139.165.158.2 ; 139.165.158.3 ; ... ; 139.165.158.100 } que l'on peut attribuer aux ordinateurs réservés aux bibliothécaires.

Traduit en nom d'hôtes complet cela nous donnera quelque chose comme :

postetrav1.biblio.organisation.org

postetrav2.biblio.organisation.org

postetrav3.biblio.organisation.org

.....

⁷Pour d'avantages d'informations, voir la directive ScriptAlias au paragraphe B.1.1.1 en annexe

postetrav100.biblio.organisation.org

Apache est un serveur Web sûr et robuste qui a pour vocation d'être sécurisé par lui-même et exposé à internet sans grand danger.

Limiter l'accès à l'interface d'administration via une directive d'Apache me semble être une mesure de sécurité suffisante.

3.1.4 la sécurité du serveur

Nous allons aborder ici la problématique de la sécurité du serveur.

Pour dissiper toute confusion, je parle ici du serveur dans son ensemble, c'est à dire, le matériel et l'ensemble des logiciels, (système d'exploitation compris) nécessaire au fonctionnement de Koha.

Nous avons vu au paragraphe 3.1.3 qu'il est possible d'améliorer la sécurité de Koha en limitant l'accès à l'interface bibliothécaire.

On peut aussi doubler cette mesure de sécurité par la configuration d'un pare-feu.

Un pare-feu est un logiciel s'exécutant sur un ordinateur qui s'interpose entre un réseau et un autre. Comme un poste de douane situé sur une route entre deux pays. Il a la charge de filtrer le trafic réseau qui transite par cet ordinateur. Ce filtrage s'exprime par des règles. Sans entrer dans les détails syntaxiques, on peut déterminer des règles qui disent ceci (traduit ebaguageo langage) :

```
Accepter tout trafic réseau venant des 99 ordinateurs du sous domaine biblio
et dont la destination est l'ordinateur sur lequel tourne Koha sur le port
8082
```

```
Accepter tout trafic réseau dont la destination est l'ordinateur sur lequel
tourne Koha sur le port 80
```

```
Accepter ceci
```

```
Accepter cela
```

```
Tout ce qui n'a pas été accepté est refusé
```

La première règle rend l'interface bibliothécaire accessible uniquement de l'intérieur du réseau de la bibliothèque. Comme nous l'avons vu à la page 9, ce qui permet de différencier l'interface bibliothécaire de l'OPAC est le numéro de port.

La deuxième rend l'OPAC accessible (de n'importe où, pour les personnes qui voudraient le consulter à partir d'internet).

Il y a une autre politique de configuration de pare feux, plus tolérante, qui consiste à tout accepter et à refuser explicitement ce qui doit l'être.

On peut ajouter le même genre de règles de pare feu pour autoriser l'accès à l'OPAC uniquement aux ordinateurs en libre accès dans la bibliothèque. Et on obtient quelque chose de terriblement restrictif mais parfaitement sécurisé.

L'inconvénient est qu'on ne peut alors accéder à l'OPAC qu'en utilisant les ordinateurs autorisés.

L'OPAC ne sera plus accessible via internet, ce qui risque de poser de sérieux problèmes et de défavoriser les étudiants ne vivant pas à proximité.

Il faut prendre conscience qu'aucune barrière n'est infranchissable et que c'est en multipliant celles-ci que l'on décourage les tentatives d'intrusions.

Il y a aussi la sécurité du matériel (local, accès physique à la machine, panne matériel). Les accès aux ordinateurs sur lesquels travaillent les bibliothécaires devront eux même être sécurisés, bien qu'il ne font pas directement partie de la problématique de la sécurité du serveur.

Quelque soit le système d'exploitation, le compte de chaque utilisateur doit être protégé par un mot de passe. Il est préférable de fermer sa session quand on quitte son poste de travail, etc. Si une personne mal intentionnée ou maladroite utilise une session restée ouverte, elle peut accéder à l'interface d'administration en utilisant les mots de passe mémorisés dans le navigateur. Il est donc préférable de ne pas laisser le navigateur enregistrer le mot de passe et de le retaper à chaque fois.

La qualité des mots de passe faciles à retenir mais difficiles à deviner est aussi un point très important. Une excellente façon de créer un mot de passe de qualité est de mémoriser un paragraphe de quelques lignes de son livre de chevet du moment, la citation d'un poème ou quoi que ce soit que l'on puisse retenir aisément. Ensuite on prend la première lettre de chaque mot en respectant majuscule et minuscule. Au moment de taper son mot de passe, il suffit de réciter mentalement ces quelques lignes, en tapant au clavier la première lettre de chaque mot.

Exemple :

Puisque ta voix étrange
 Vision qui dérange
 Et trouble l'horizon
 De ma raison

Donne comme mot de passe : PtvéVqdEt1'hDmr

En conclusion :

limiter l'accès à l'interface d'administration via une directive d'Apache et ne pas utiliser les règles de pare feu précédemment citées, me semble être une mesure de sécurité suffisante. Il est préférable de ne pas utiliser de règle interdisant l'accès à l'OPAC aux ordinateurs ne faisant pas partie de réseau. Ainsi l'OPAC reste réellement publique et accessible de n'importe où.

On peut aussi faire tourner un pare-feu sur le serveur⁸ lui même, mais il sera toujours temps de s'en donner la peine en cas de problème.

3.2 Koha sous Windows

Au niveau applicatif, la sécurité est identique quelque soit l'OS⁹, mais une perte de données peut être simplement causée par un plantage du système dans une situation d'incohérence de la base de données.

Un piratage peut s'effectuer directement via un éventuel trou de sécurité du système d'ex-

⁸On utilise le mot serveur aussi bien pour désigner un service s'exécutant sur un ordinateur comme Apache, MySQL, etc, que pour désigner l'ordinateur lui même. Dans ce cas je parle de l'ordinateur.

⁹Operating System (système d'exploitation)

plotation sans qu'il ne soit nécessaire de "sniffer" les mots de passe transitant sur l'intranet.

Et c'est bien là que Windows est défaillant.

Il n'est pas normal qu'un script VBscript (Visual Basic Script) simplement attaché à un courriel puisse s'exécuter. L'API (Application Programming Interface) VBScript est présente sur la plupart des logiciels Microsoft, le problème c'est qu'un simple script peut prendre le contrôle, ou endommager une partie du système d'exploitation.

Le VBscript est un langage limité à Windows. Les virus VBscript sont donc des virus Windows limités à Windows (malheureusement très efficaces dans ce domaine).

Le moins qu'on puisse dire c'est que ce n'est pas la bonne façon d'implémenter la possibilité de personnaliser un logiciel ou d'étendre ses fonctionnalités. A cause de cela, certains virus qui n'étaient que du simple VBscript volontairement malveillants ont fait des ravages à travers le monde. Ils ont paralysé de nombreux systèmes informatiques de première importance, basés à tort sur un OS windows... Par incompetence des décideurs ?

Même si Koha peut fonctionner sous Windows, je déconseille donc de le faire. Windows exige trop d'intervention pour le protéger des virus, les éradiquer, réinstaller le système lorsqu'il devient instable, etc.

Windows n'est pas un système suffisamment sûr et stable pour lui confier le bon fonctionnement du serveur. De plus, si l'on veut sécuriser Windows, il faudra acheter les licences du système, et des logiciels de sécurité.

http://pricelist.checkpoint.com/sections/Enterprise_sku.asp?prod_NAME=36

Même si windows paraît plus simple à comprendre au premier abord, il vaut mieux installer Koha sur un système GNU/Linux ou BSD¹⁰.

3.3 PMB sous Linux

Ne disposant pas de documentation sur le développement de PMB je n'ai pu mener à bien un étude détaillée de sa sécurité.

Pour étudier la sécurité de Koha, j'ai lu la documentation destinée aux développeur et sur base de ces informations, j'ai pu consulter le code source. L'analyse du code source de PMB sans aucune information de départ aurait été trop hasardeuse, et aurait pris trop de temps.

L'aspect sécurité de PMB a énormément de points communs avec celui de Koha. Seul diffère le langage utilisé et bien sûr, la manière dont la sécurité est développée. Les informations des paragraphes 3.1.2.1, 3.1.2.4, 3.1.3 et 3.1.4 sont valables pour PMB.

L'installation par défaut a configuré PMB pour que l'OPAC soit sur le même port que l'interface bibliothécaire. C'est bien dommage car, dans ce cas, il est impossible de faire (par des règles de pare feu) deux politiques de sécurité différentes. Cette lacune de sécurité serait-elle destinée à assurer la compatibilité avec les serveurs IIS¹¹ ? Pour que les deux interfaces tournent sur des ports différents, il faut le faire en chargeant les directives appropriées. Différentes entre les serveurs.

¹⁰FreeBSD <http://www.freebsd.org/>
OpenBSD <http://www.openbsd.org/>
NetBSD <http://www.netbsd.org/>

¹¹IIS est le serveur WEB de microsoft

3.4 PMB sous Windows

La sécurité de PMB sous Windows pose les même problèmes qu'avec Koha (Cf. ??).

Chapitre 4

Import/export des données

4.1 Facilité d'importation des notices dans tous les formats standards (MARC, ISO, etc)

Tout bon logiciel et à plus forte raison un logiciel Open source doit inclure toutes les fonctionnalités nécessaires à l'échange de données vers et en provenance d'autres programmes. La compatibilité des formats d'échange de fichiers avec d'autres programmes est essentielle afin de garantir une des libertés fondamentales qu'est le droit de choisir son logiciel.

Certains éditeurs de logiciels propriétaires ne respectent pas ce droit. Le format des documents qu'ils utilisent dévie de la norme. D'autres n'offrent aucune possibilité d'importation ou d'exportation dans un format normalisé.

L'objectif étant de maintenir l'utilisateur captif, car si il veut changer de logiciel, il perdra une partie du travail réalisé.

Si l'on prend comme exemple les outils bureautiques, il existe une norme pour le format des documents qu'est l'OpenDocument.

http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office

<http://www.oasis-open.org/committees/download.php/12572/OpenDocument-v1.0-os.pdf>

Cette norme a été établie par L'OASIS (Organization for the Advancement of Structured Information Standards) qui est un consortium mondial (dont font partie Microsoft, Adobe, Oracle, Sun Microsystems, etc) qui travaille pour la normalisation et la standardisation de formats de fichiers ouverts basés sur XML.

Les logiciels utilisant un système de base de données relationnel (comme Koha et PMB) posent rarement des problèmes d'échange de données... Même si le logiciel ne prévoit aucune fonction d'importation ou d'exportation, la base de données le fera.

Les échanges de données entre les SIGB se font dans un format de fichier standardisé qu'est l'ISO2709.

Koha inclut une fonction d'importation/exportation de notices. Ces notices sont stockées dans un fichier au format respectant le standard ISO2709. L'importation peut aussi se faire à partir d'un réservoir auquel on accède grâce à un client Z3950.

Ces fonctionnalités d'import-export de données sont indispensable au bon usage d'un SIGB, et ce pour des raisons évidentes d'échange, de collecte et de centralisation de la connaissance

dans une bibliothèque.

Le partage de la connaissance est aussi intimement lié à l'éthique du logiciel libre. C'est entre autre sur cette base qu'il a pris le formidable essor qu'on lui connaît.

4.1.1 Importation des données provenant de CDS/ISIS

De nombreuses tentatives de récupération des données provenant de CDS/ISIS ont été effectuées, elles n'ont pas encore abouti à des résultats totalement satisfaisants bien qu'il soit théoriquement possible d'exporter à partir de CDS/ISIS ou WINISIS vers un fichier ISO2709. Cependant, le problème principal réside dans le fait que la structure des données de CDS/ISIS est entièrement paramétrée par l'utilisateur. Dès lors, les catalogues créés avec CDS/ISIS utilisent généralement un format de catalogage spécifique à la bibliothèque qui ne correspond pas à un standard MARC.

L'importation de données en provenance de CDS/ISIS doit donc se faire à l'aide d'une table de correspondance permettant de lier un champ paramétré de CDS/ISIS avec le champ et sous-champ correspondant au standard du MARC utilisé.

Voici les liens et pistes qui ont été suivis

- Une question a été posée sur une liste de diffusion de CDS/ISIS¹ concernant les possibilités offertes par CDS/ISIS d'exporter les données bibliographiques vers un fichier ISO2709 au format MARC.

Aucune suggestion des abonnés de cette liste n'a permis de parvenir à ce résultat.

Cf. http://www.raymond-gimilio.fr/article.php3?id_article=7

- La même question a été posée sur la liste de diffusion de Koha. Un abonné de cette liste a répondu par l'envoi de quelques explications ainsi que ce lien :

<http://dlieppe.free.fr/ExportWinISIS/ExportWinISIS-ISO2709.htm>

Toutefois l'usage du logiciel recommandé sur ce document renvoie au problème de l'absence de table de conversion permettant de lier les champs de CDS/ISIS avec les tags et sous champs du MARC. Le fichier ISO2709 ainsi exporté ne peut donc pas assurer la cohérence des données.

- Une recherche sur google renvoie les liens suivant :

- <http://www.unesco.org/isis/files/winisis/windows/doc/francais/>

- <http://www.unesco.org/isis/files/winisis/windows/utilities/>

- http://www.saas.nsw.edu.au/koha_wiki/index.php?page=CDS%2FISIS+based+system+to+Koha

- <http://216.239.59.104/search?q=cache:6vnLVmbThY4J:lists.katipo.co.nz/public/koha/2004/003351.html+marc21+winisis+koha&hl=fr>

- <http://recifweb.free.fr/main.php?page=edito>

- http://portal.unesco.org/ci/admin/ev.php?URL_ID=11041&URL_DO=DO_TOPIC&URL_SECTION=201&reload=1106836908

Une piste prometteuse a été étudiée à l'intérieur même de CDS/ISIS. Il est en effet possible possible en CDS/ISIS et WINISIS d'utiliser une table de reformatage lors de l'exportation ISO2709 de manière à redistribuer les données autrement, selon les tags MARC21 par

¹afucientraide@yahoogroupes.fr

exemple. Des essais ont dès lors été faits en ce sens à partir d'un échantillon de données issues du catalogue de l'UNR.

Une première difficulté est liée au fait que le contenu des champs de ce catalogue ne peut pas directement être mis en correspondance ("matché") avec des champs et sous-champs MARC, la structure d'analyse de l'information étant différente (par exemple mentions de responsabilité...). Cependant, il semble d'après les bibliothécaires qu'un degré suffisant d'équivalence peut quand même être obtenu moyennant parfois quelques opérations lors de la préparation de ce fichier de reformatage.

Malheureusement, il s'avère que le fichier ISO2709 obtenu de cette façon ne respecte pas parfaitement la structure standard d'un fichier ISO2709, notamment au niveau du contenu du répertoire. Malgré divers essais en modifiant la configuration de l'exportation, aucune opération d'importation parfaitement satisfaisante dans Koha n'a pu être réalisée. Ceci s'explique peut-être par notre manque de connaissance dans le langage de formatage spécifique de CDS/ISIS. Par manque de temps, il ne nous a pas été possible d'aller plus loin dans ces essais. Cette piste mérite néanmoins d'être approfondie.

Cette absence de solution conviviale, n'est pas un critère disqualifiant pour Koha, mais plutôt pour le logiciel CDS/ISIS. Par son modèle de développement propriétaire et son non respect des standards de structures de données bibliographique, CDS/ISIS rend ses utilisateurs captifs et les place face à une difficulté importante lorsqu'ils souhaitent migrer vers d'autres logiciels.

Seuls les logiciels offrant une possibilité d'importation des données de CDS/ISIS avec une table de correspondance des données permettront d'assurer facilement cette migration.

Pour les logiciels de gestion bibliographique tels que Koha qui ne disposent pas de module spécifique à l'importation de données provenant de CDS/ISIS, il reste possible d'effectuer cette migration manuellement. Mais cette migration nécessite des compétences informatiques qui pourraient être investies dans le développement d'un module d'acquisition disposant d'une table de conversion.

Il existe des logiciels payants permettant d'assurer l'acquisition de notices tels que :

ZMarcCollector (350\$) qui permet de se connecter à des réservoirs en utilisant un client Z3950. Ce logiciel se base sur un fichier texte contenant une liste d'ISBN. Les données ainsi collectées peuvent être exportées dans un fichier ISO 2709.

<http://www.ait.com.au/>

BIBLIObase permet de faire des exportations dans un fichier ISO 2709.

<http://www.bibliosoft.pt/>

BookWhere permet de faire une recherche simultanée sur 1900 réservoirs prédéfinis et ensuite d'exporter le résultat en MARC21 ou UNIMARC.

<http://www.webclarity.info/>

D'autres logiciels fonctionnant sous DOS² sont proposés par la Library of Congress pour assurer la migration des données. Mais ils ne fonctionnent pas avec les versions du DOS

²<http://www.loc.gov/marc/makrbrkr.html>

ultérieures à Windows 95. Des tests ont été faits avec un émulateur de cette antique version du DOS (dosbox) mais sans succès.

Il existe également une documentation sur le site de Koha qui explique comment migrer des données provenant de CDS/ISIS mais la solution proposée ne convient pas dans la mesure où les scripts qui y sont présentés ne proposent aucun moyen de faire correspondre les champs définis par les utilisateurs de CDS/ISIS avec les champs du MARC.

Enfin un logiciel prometteur³ a été trouvé. Il permettrait de convertir un fichier texte délimité par une tabulation (CSV) en fichier ISO2709. Malheureusement CDS/ISIS n'offre pas de possibilité d'exporter directement sa base de données vers un fichier CSV. On pourrait toutefois y parvenir par un travail de formatage de l'impression dans un fichier ASCII (Dans CDS/ISIS : Fichier=>imprimer=>Sortie vers. Ensuite définir le format de sortie pour chaque champ dans la zone d'édition). Cette solution ne peut pas être considérée comme idéale mais pourrait être envisagée.

Si la piste de reformatage et exportation à l'intérieur de CDS/ISIS évoquée ci-dessus n'aboutit pas, un logiciel présentant une table de conversion devra être développé pour assurer le succès de la migration des données provenant de CDS/ISIS vers Koha.

J'ai rédigé une note destinée aux informaticiens qui donne un exemple de méthode d'implémentation d'un tel logiciel. Cette note se trouve en annexe au paragraphe E.3.3. Il ne s'agit pas d'une analyse, cette note ne permet pas d'évaluer le temps nécessaire à l'étude (lecture de documentations), l'analyse, le développement et le test d'un tel logiciel.

J'ai essayé de développer ce programme en Perl avec l'API (Application Programming Interface) SAX. Faute de temps, et d'une bonne documentation sur l'usage de SAX en Perl, ce travail n'a pas pu être achevé.

<http://search.cpan.org/~msergeant/XML-SAX-0.12/SAX/ParserFactory.pm>

<http://search.cpan.org/~msergeant/XML-SAX-0.12/SAX/Intro.pod>

<http://search.cpan.org/~msergeant/XML-SAX-0.12/SAX/PurePerl.pm>

L'API SAX en Java est parfaitement documentée, mais la description de son utilisation en Perl laisse à désirer.

4.2 Facilité d'exportation des notices dans tous les formats standards (MARC, ISO, etc)

Koha inclut une fonction d'exportation de notices au format ISO2709.

Paramètres => MARC biblio export

Il suffit de fournir deux intervalles de la notice n à n' ou de ne rien indiquer pour exporter l'ensemble des notices.

Cette opération s'est déroulée sans problème lors des tests. Les enregistrements sont chargés sur le poste de travail qui a demandé l'exportation et sont sauvés dans un fichier nommé "marc.pl".

³<http://oregonstate.edu/~reerset/marcredit/html/index.html>

Chapitre 5

Qualité des aides

Cette partie du document propose un aperçu des sources d'information disponibles sur Koha. Ces sources d'informations sont :

- le site de la documentation officielle de Koha ;
- les listes de diffusion ;
- l'aide en ligne intégrée au logiciel.

À la fin de ce chapitre (au paragraphe 5.4) quelques lignes seront consacrées à une critique succincte de la documentation de PMB.

5.1 Le site de la documentation officielle de Koha ;

Toute la documentation sur Koha est centralisée à l'adresse suivante :

<http://kohadocs.org/>

L'arborescence de cette page se présente comme suit :

1. Les informations générales sur Koha comportant :
 - un guide démarrage pour débutant(en français). Ce document est rédigé dans un style agréable avec quelques touches d'humour, mais il n'est pas très instructif.
http://www.kohadocs.org/guide_demarrage.html
 - Un nouveau manuel de l'utilisateur en cours de rédaction (en anglais) vient d'être ajouté au site.
A ce jour, il n'existe toujours pas de manuel complet sur Koha. Aucun pronostic ne peut être fait quand à une traduction en français.
Si l'on s'en tient à son manuel, où plutôt à son absence, Koha est un des logiciels les plus mal documentés qui soit.
Le premier chapitre a été publié le 7 mai 2005, et est disponible à cette adresse.
<http://www.kohadocs.org/usersguide>
Cette documentation, me paraît être d'excellente qualité d'après le premier chapitre, cette absence de documentation complète sera ainsi solutionnée dès que le document sera entièrement rédigé.
- (a) Vous trouverez aussi une liste des fonctionnalités de la version courante de Koha.
http://www.kohadocs.org/Description2_2.html

2. L'administration système :

- (a) Les guides d'installation de koha :

- L’installation de Koha sur une Debian GNU/Linux : Ce guide donne de précieuses informations sur l’installation des logiciels de base et des bibliothèques Perl nécessaires à Koha. Toutefois il est incomplet et ne fournit pas toutes les réponses aux questions posées par l’assistant d’installation des bibliothèques Perl. Il omet certaines bibliothèques pourtant requises par Koha. Enfin le passage de la version stable de Debian à Sarge rend ce document obsolète.
http://www.kohadocs.org/Installing_Koha_on_Debian_Woody.html
C’est la raison pour laquelle j’ai traduit librement ce document en français, ai ajouté des informations pour le rendre plus exhaustif et ai mis à jour les informations obsolètes. Vous trouverez ce document en annexe D à la page 91
 - l’installation de Koha sur la distribution GNU/Linux Fedora.
http://www.kohadocs.org/Installing_Koha_on_Fedora.html
 - Un autre sous windows.
<http://koha.rwjr.com/>
- (b) Migration des données.
- Migration vers Koha :
Ce document explique comment importer dans Koha des données venant d’un autre SIGB. Il explique la structure des données de Koha. Et comment importer un fichier texte au format csv (comma-separated value) dans chaque table de la base de données de Koha. Le problème est que chaque champ de ce fichier csv doit être d’une structure identique à la table que l’on désire alimenter, ce qui implique une opération initiale de préformatage qui peut être complexe.
http://www.kohadocs.org/migrating_to_koha.html
 - Migrer des données à partir de CDS/ISIS (Cf. paragraphe 4.1.1) :
La solution proposée ne convient pas dans la mesure où les scripts qui y sont présentés ne proposent aucun moyen de faire correspondre les champs définis par les utilisateurs de CDS/ISIS avec les champs du MARC.
http://www.kohadocs.org/CDS_ISIS_to_Koha.html
 - Une documentation en français décrivant la migration des données à partir d’un fichier iso2709 au format MARC.
http://www.kohadocs.org/migrer_ses_donnees.html
- (c) Templates.
- Deux documents, l’un en anglais, l’autre en espagnol sur la traduction des templates (Cf. 2.3.1) et les outils permettant d’internationaliser ceux-ci.
<http://www.kohadocs.org/KOHATranslation/translation.html>
- (d) Maintenance et résolution de problèmes.
- Cette partie de la documentation traite des sujets suivants :
- La réalisation des mises à jour de Koha.
http://www.kohadocs.org/Updating_Koha.html
 - Les sauvegardes et la restauration de celles-ci.
http://www.kohadocs.org/Backups_and_Restoring.html
Le même sujet est abordé au paragraphe 3.1.2 relatif à la sécurité des données. Dans le cadre de l’étude des possibilités de sauvegardes des données, j’ai rédigé un manuel couvrant toutes les possibilités de sauvegarde et de restauration liées à MySQL. Vous trouverez ce document en annexe au paragraphe C.1.
 - Scripts ligne de commande.
Document en français expliquant le rôle de chaque script en ligne de commande fourni avec l’installation de Koha.

http://www.kohadocs.org/scripts_commande.html

3. Online Public Access Catalog (OPAC)

Le manuel en français de l'OPAC de Koha.

<http://www.kohadocs.org/CatKOHA/CatKOHA.html>

4. Développement de Koha.

Excellent document comportant notamment un modèle des données de la base de données de Koha. Il explique très succinctement le modèle de développement de Koha et sa répartition en paquetages et modules.

http://www.kohadocs.org/Koha_architecture.html

Ce document est destiné aux informaticiens.

5.1.1 Conclusion :

La documentation de Koha n'est pas très bien structurée, il n'existe pas de manuel complet de Koha¹, excepté celui de l'OPAC.

Certains documents présentent des listings de sources de scripts dans le contenu de la page au lieu de proposer un lien permettant de télécharger le fichier. En plus d'avoir peu d'intérêt pour la plupart des lecteurs, ceci oblige le programmeur à avoir recours à de grands copier coller de la page, avec le risque de chevauchement ou d'omission de certaines lignes.

Globalement, la documentation de Koha laisse à désirer, raison pour laquelle j'ai entrepris de rédiger certains documents sur des sujets qui traitent de la partie purement informatique de Koha.

5.2 Les listes de diffusion

Elles sont extrêmement réactives, se composent d'une liste en français² et d'une autre en anglais³.

Koha jouissant d'un très importante communauté d'utilisateurs et de développeurs fait que toute question postée sur l'une de ces liste obtient une réponse dans l'heure, voir même après dix minutes. J'y ai bien sûr eu recours dans le cadre de ce travail et les réponses obtenues ont la plupart du temps été correctes et utiles.

Ces listes extrêmement dynamiques viennent compenser la documentation de base insuffisante.

5.3 Aide en ligne

Les aides en ligne sont les aides contextuelles que l'on peut obtenir en "cliquant" sur "Aide" à droite dans le menu de Koha. Elles sont globalement de bonne qualité mais certaines fonctionnalités en sont dépourvues. Par exemple la fonction d'ajout de notice ne comporte pas d'aide. Il faut donc deviner comment on se sert de la recherche sur un réservoir distant avec un client Z39.50, comment lier la notice à un exemplaire, etc. En fin de compte, une des fonctions de catalogage des plus importantes d'un SIGB ne comporte pas d'aide.

¹Le guide de l'utilisateur de Koha est en cours de rédaction. Cf. <http://www.kohadocs.org/usersguide/>

²infos@koha-fr.org

³koha@lists.katipo.co.nz

En conclusion, le guide de l'utilisateur de Koha constituera une aide précieuse lorsqu'il sera terminé. En attendant, le meilleur moyen d'obtenir une information reste les listes de diffusions d'un extraordinaire dynamisme. Quand à la documentation existante, les informations y sont éparpillées dans une foule de documents parfois obsolètes et incomplets.

5.4 La documentation de PMB

L'objectif de ce paragraphe est d'apporter une critique succincte de la qualité de la documentation, ses points faibles, ses point forts et ce qui pourrait être fait pour l'améliorer. Une critique comparative par rapport à Koha m'a semblé être une démarche intéressante, il ne faut en aucun cas y voir une quelconque forme de provocation⁴.

En premier lieux, il faut en saluer l'existence. La documentation de PMB est répartie en trois documents :

1. Installation.

<http://www.sigb.net/doc/html-install/index.html>

Elle est décrite au paragraphe 1.3.1 du chapitre 1.

2. Utilisateur.

<http://www.sigb.net/doc/html-user/index.html>

3. Administrateur.

<http://www.sigb.net/doc/html-admin/index.html>

Contrairement à la documentation de Koha, les 3 documents en français sont bien présentés dans un langage clair, avec des saisies d'écran pour chaque fonctionnalité décrite. Toutefois les développeurs sont laissés pour compte. Dans ce domaine, c'est la documentation de Koha qui s'illustre le plus brillamment.

Raison pour laquelle, j'ai pu décrire en détail le modèle de développement de Koha et pas celui de PMB.

⁴Appelée troll dans le jargon des listes de diffusion informatique.

Chapitre 6

Possibilité d'évolution

6.1 Évolutions à court terme prévues par l'équipe de développement de Koha

Le projet de développement de Koha a été initialisé fin 1999, en Nouvelle Zélande. C'est un logiciel dont le développement jouit du dynamisme d'un très grand nombre de développeurs de par le monde. Il est en constante évolution. La 1ère version est sortie en janvier 2000, la deuxième début 2002 et nous sommes actuellement à la version 2.2.3. Le projet a commencé à recevoir des investissements de sociétés commerciales à partir de fin 2002.

Toute personne qui le désire peut participer au développement de Koha, ainsi on peut télécharger et utiliser librement l'application mais aussi la modifier puisqu'on dispose des codes sources¹.

Le code source est composé de lettres et de chiffres, de mots et de phrases compréhensibles par le programmeur. Il est rédigé en respectant une grammaire et une syntaxe propre au langage utilisé, Perl en l'occurrence. Ce code est ensuite interprété² par un logiciel lié au langage utilisé afin de le rendre compréhensible par l'ordinateur qui l'exécute.

Il y a deux façons d'améliorer ou de modifier Koha :

- Télécharger le code source, le modifier et l'exécuter sur un ordinateur.

Mais dans ce cas les modifications ne seront effectives que sur cet ordinateur.

- Faire partie de l'équipe de développement.

Dans ce cas tout le monde télécharge et profite des modifications qui sont apportées au code source.

Le développeur est bien sûr supervisé par un responsable, soit d'une spécificité soit régional. Son travail est donc contrôlé avant de faire partie du code proposé au téléchargement. Dans les grands projets de développement libre comme celui de Koha il y a aussi des volontaires beta testeur qui testent les nouvelles versions du logiciel. **Il y a peu de risque qu'un code erroné soit distribué à grande échelle. Et même si une erreur passait inaperçue dans un premier temps, elle serait rapidement signalée aux développeurs.** En effet, le code sera relu par un grand nombre de personnes puisqu'il est disponible.

¹C'est une des libertés fondamentale qui fait la spécificité du logiciel libre.

²Perl qui est le langage de programmation de Koha est interprété à la volée. Mais il existe aussi des langages qui nécessitent une compilation préalable comme le C, C++, etc.

Généralement, les choix stratégiques de développement sont pris en commun par l'équipe de responsables et développeurs au gré de l'évolution des besoins des utilisateurs, des systèmes, de nouveaux problèmes de sécurité, etc³.

A titre d'exemple, voici quelques projets d'évolution à court terme de Koha⁴ :

6.1.1 Gestion multi-site améliorée

Les budgets pourront être répartis par sites

6.1.1.1 Budget

En fonction d'un budget affecté à un site, les bibliothécaires de différents sites pourront ou non faire des acquisitions sur ce budget. Celui-ci pourra choisir le budget dans une liste restreinte.

6.1.1.2 Site distinct

Pour répondre aux besoins de certaines bibliothèques multi-site dont chaque site dispose de sa propre équipe de catalogage et d'acquisition, l'activation d'une option pourra modifier le comportement du module acquisition de la façon suivante :

- dans le module acquisition :
 - seules les suggestions faites par un utilisateur du même site que le bibliothécaire peuvent être acceptées/rejetées ;
 - une commande peut être modifiée/reçue/fermée seulement par un bibliothécaire venant du même site que celui qui a créé le panier.
- dans le module catalogue, une notice peut être modifiée/effacée/créée seulement par un bibliothécaire du site propriétaire du document ;
- dans le module lecteur, un lecteur peut être créé/modifié seulement par un bibliothécaire du même site que le lecteur.

6.1.2 Autres améliorations

6.1.2.1 Tracer qui fait quoi

Koha va enregistrer quel bibliothécaire crée/modifie une notice, et quand.

6.1.2.2 Commandes en retard

Dans les acquisitions, créer une nouvelle page pour voir, pour un libraire donné, les lignes des commandes qui ne sont toujours pas reçues.

³Je n'ai pas eu le temps de m'informer sur les règles exactes régissant la politique de développement de Koha, mais c'est en gros comme ça que ça se passe la plupart du temps dans les modèles de développement libre.

⁴Comme souvent dans les projets de développement libre où la qualité passe avant le marketing, aucune date n'est fixée. Les annonces sur les listes de diffusion parlent en terme de bientôt, dès que possible...

6.1.2.3 Traceur de l'historique du libraire

Un outil pour suivre l'historique des relances (pour les acquisitions et les périodiques). Quand le bibliothécaire réclame un livre ou un périodique en retard au fournisseur, la réclamation sera stockée dans une table historique.

6.1.2.4 Liste de circulation

Dans les périodiques (création d'abonnement), le bibliothécaire pourra sélectionner des utilisateurs recevant les numéros d'un périodique à tour de rôle.

6.1.2.5 Alerte de parution d'un périodique

Avec cette fonctionnalité, pour les périodiques, un utilisateur peut s'abonner à "une alerte de parution". Pour tous les volumes déclarés arrivés/manquants, un mail est envoyé à tous les abonnés de cette liste. Le mail prévient l'utilisateur que le volume est arrivé ou manquant.

6.1.2.6 Calcul de la date de retour

La date de retour permettra de tenir compte de l'échéance de l'inscription de l'emprunteur.

6.1.3 Conclusion

Cette liste ne concerne que les évolutions à court terme, Koha promet de belles perspectives de développement futur tant son dynamisme est remarquable. Une page d'informations sur la prochaine version stable de Koha (2.2.4) est disponible à cette adresse :

<http://wiki.liblime.com/doku.php?id=koha24rmnotes>

6.2 Possibilité d'ajout de fonctionnalités à Koha

Dans cette partie, nous allons voir dans quelles mesures il est possible de modifier le code source de Koha de manière durable sans faire partie de l'équipe de développement.

Les principaux problèmes qui se posent :

- les modifications doivent respecter le modèle de développement de Koha ;
- elles doivent être compatibles avec les mises à jour du code proposées par les développeurs officiels de Koha. Ces mises à jour officielles ne tenant pas compte des modifications apportées localement au code.

Nous verrons ensuite au paragraphe 6.2.1 comment s'effectue une mise à jour officielle et ce qu'il faudra faire pour préserver les modifications localement apportées au codes, ou comment s'assurer que ces modifications soient aisément remises en place si elles sont écrasées par une mise à jour de Koha.

Pour mieux comprendre cette partie du document, il est préférable d'avoir lu le paragraphe 2.3

Cette partie du document est assez technique, puisqu'elle traite d'un sujet rébarbatif qu'est la programmation. Elle a toutefois été rédigée de façon à pouvoir être lue par une personne

n'ayant aucune notion de programmation⁵.

Si l'on devait développer de nouvelles fonctionnalités pour Koha, ce serait très aisé à condition que le lien (URL⁶) pointe vers une page séparée. Dans ce cas on ajoute un fichier script au code source de Koha mais on ne modifie pas le code existant. Il n'y a alors pas ou peu de problèmes d'interaction avec le code existant. Nous avons vu au paragraphe 2.3 que le code de Koha est partagé en modules. Si l'on désirait ajouter une fonctionnalité à Koha, il faudrait aussi lui ajouter des modules.

Le problème qui se pose alors est que les nouvelles fonctions principales⁷ trouvent les modules ajoutés au code existant.

Pour ajouter un nouveau module il faut soit :

- modifier le fichier de configuration de Koha ;

Pour qu'il prenne en compte le chemin sur le système de fichier vers le nouveau module, il faut changer la variable PERL5LIB et y ajouter un répertoire séparé par deux points " :"

ex :

```
SetEnv PERL5LIB "/koha/intranet/modules :/autre/chemin"
```

De cette façon les mises à jour de Koha n'écraseront pas le répertoire des modules, mais il serait préférable de les placer dans un sous répertoire de /koha/intranet/modules, et de les recopier lors d'une mise à jour.

- spécifier le chemin complet du module dans chaque code source où ce module est requis.

utiliser la syntaxe :

```
use lib "/my/directory";  
dans le code des scripts ajoutés.
```

Si on désire intégrer le lien vers de nouvelles fonctionnalités dans une page existante, il suffit de modifier le template⁸ de la page html. Il n'est pas nécessaire de modifier un script existant.

Je n'ai pas eu l'occasion de tester le développement et l'ajout de nouveaux modules. Un travail d'une telle ampleur ne se justifie que si Koha ne répond pas complètement aux besoins des utilisateurs. Il dépasse le cadre de cette étude.

6.2.1 Comment effectuer une mise à jour de Koha après avoir modifié son code source

Lorsque l'on exécute le script "koha.upgrade" pour effectuer une mise à jour de Koha, celui-ci renomme les répertoires :

- cgi-bin qui contient les scripts

⁵Les parties de ce document encadrées ne sont pas nécessaires à la compréhension du texte, elles constituent un complément d'information destiné aux informaticiens.

⁶voir définition à la page 102 en annexe

⁷Les fonctions principales sont celles qui sont appelées lorsque l'on clique sur un lien. Ces fonctions principales font appel à des sous fonctions placées dans les modules.

⁸voir paragraphe 2.3.1.1

- htdocs qui contient les templates
 - modules qui contient les modules
 - scripts qui contient le démon z3950 et quelques autres scripts divers.
- Crée de nouveaux répertoires du même nom et y copie ses nouveaux fichiers.

Lors de chaque mise à jour, les ajouts de fonctionnalités seront perdues.

Le développeur devra soit proposer l'ajout de ses fonctionnalités au projet Koha, ou bien, pour chaque sortie d'une nouvelle version de Koha, effectuer les opérations suivantes :

- s'assurer que les templates ne sont pas modifiés par la mise à jour, si non elle écraserait l'accès aux nouvelles fonctionnalités.
- s'assurer que le fichier `/etc/koha-httpd.conf` n'a pas été écrasé
- rédiger un script qui recopie :
 - le template nouvellement développé ou l'ancien (voir pt précédent) dans le répertoire htdocs
 - les scripts contenant les fonctionnalités ajoutées dans cgi-bin
 - Recopier les modules à leur place...

6.2.2 Conclusion

Si le développeur :

- se contente d'ajouter un lien vers une page séparée dans le template
- qu'il n'utilise aucun module de Koha pour développer ses fonctionnalités,
- qu'il n'est pas nécessaire de les intégrer dans une page ou une fonction existante,

le travail de maintenance ne représente que quelques heures pour chaque sortie d'une nouvelle version de Koha. Au contraire, si le développeur désire étendre ou modifier une fonctionnalité existante alors les choses se compliquent.

Si les modifications ne sont pas ajoutées au projet officiel de Koha, le développeur des nouvelles fonctionnalités devra en assurer la maintenance. Mais comme nous l'avons vu précédemment, il est aussi possible de faire partie de l'équipe de développement de Koha, car c'est la philosophie du logiciel libre.

Même si le programmeur des nouvelles fonctionnalités ne fait pas partie de l'équipe de Koha, il peut aussi lui proposer d'intégrer ses modifications à la version officielle.

Enfin la plupart des projets de modifications cités aux paragraphes 6.1.1 et 6.1.2 proviennent de nouveaux besoins formulés par la communauté d'utilisateurs. Il est donc tout à fait possible d'exprimer un tel besoin et qu'il soit pris en compte dans une version ultérieure de Koha.

6.3 Possibilité d'évolution de PMB

PMB est un logiciel libre au même titre que Koha. Ceci implique que sa pérennité est assurée. Même si la société qui développe Koha devait disparaître.

A partir du moment où le code appartient au domaine public, tout qui le désire, peut entreprendre d'assurer la maintenance du code.

Il est préférable que le développement soit assuré par une équipe mondiale décentralisée, mais supervisée par un ou plusieurs responsables. C'est le cas pour le noyau Linux, supervisé par Linus Torvalds et la distribution Debian par Ian Murdock.

Mais de nombreux projets libres peuvent aussi se développer au sein d'une firme commerciale et acquérir la même ampleur. C'est le cas de la distribution GNU/Linux Suse <http://www.suse.de/fr/>, rachetée par Novell. Mandriva <http://www1.mandrivalinux.com/fr-be/>. Le moins que l'on puisse dire c'est que ces distributions se portent bien et jouissent d'un excellent dynamisme de développement.

Dans le code source de PMB, les commentaires sont en français, mais ce n'est pas grave, on peut les traduire facilement. Ce qui est plus problématique c'est que certaines variables et noms de fonctions (ou méthodes dans les classes) sont nommées en français, d'autres en anglais dans un même script.

Dans un projet d'une telle ampleur, il eut été souhaitable de nommer les variables et les fonctions dans une langue que tous les programmeurs du monde puissent comprendre. Dans l'état actuel des choses, un programmeur qui ne comprend pas le français ne va rien comprendre au code source. Cette situation contrevient à la plus importante des règles : la lisibilité du code par un plus grand nombre possible de personnes.

Solution :

Il faut tout traduire et effectuer du "debuggage" pour s'assurer que l'on n'a rien oublié. En effet, une variable ou une fonction mal traduite génèrerait une erreur dans le code.

Ce n'est pas un très gros travail et c'est un très bon exercice pour s'imprégner de la logique du code.

Chapitre 7

Synthèse et conclusion

7.1 Le logiciel libre

Koha et PMB sont des logiciels libres.

La caractéristique principale du logiciel libre est l’ouverture et la visibilité de son code source¹. Grâce à cela, toutes les personnes ayant des connaissances en programmation peuvent lire le code, le comprendre, l’améliorer et signaler d’éventuelles erreurs au développeur du logiciel. On obtient ainsi un code lu, relu et corrigé un nombre incalculable de fois par un nombre illimité de personnes.

Au contraire, le code source d’un logiciel propriétaire n’est connu que par un nombre restreint de programmeurs tenus au secret, contraints à la productivité et incapables de déceler leurs propres erreurs. Il est bien connu que si il est aisé de corriger les fautes d’orthographe d’un autre, on est parfois incapable de trouver les siennes. Il en va de même pour les erreurs de grammaire et de logique d’un langage de programmation. Même si généralement les équipes de programmeurs et de “debuggers” ne sont pas constituées par les mêmes personnes, le temps qui est imparti à cette tâche est limité et il est impossible de tester et relire le code autant de fois qu’il le faudrait.

Il en va de même pour les trous de sécurité. Dès qu’une faille de sécurité est détectée dans un logiciel libre, elle est rendue publique, et parfois corrigée dans les heures qui suivent.

Un logiciel propriétaire représente un coût de conception et de développement qui doit être minimisé pour optimiser le bénéfice des ventes. Les éditeurs préféreront toujours investir dans une nouvelle version de leur logiciel en prenant en compte la correction, plutôt que dans la maintenance d’un ancien produit ne générant plus de bénéfice. Certaines firmes vont jusqu’à cacher d’éventuels problèmes pour :

- éviter que les clients exigent des investissements non rentables dans la correction d’une ancienne version ;

¹Le code source est composé de lettres et de chiffres, de mots et de phrases compréhensibles par le programmeur. Il est rédigé en respectant une grammaire et une syntaxe propres au langage utilisé. Perl et Php sont, par exemple, les langages respectifs de Koha et PMB. Ce code est ensuite interprété ou compilé (Perl et Php sont des langages interprétés à la volée. Mais il existe aussi des langages qui nécessitent une compilation préalable comme le C, C++, etc.) par un logiciel lié au langage utilisé afin de le rendre compréhensible par l’ordinateur qui l’exécute.

- éviter qu’un hacker ne prenne connaissance de la faille non corrigée, l’exploite pour l’analyser, et rende publique le modus operandi afin de dénoncer la pratique commerciale de l’éditeur. Généralement le hacker propose des solutions pour sécuriser le logiciel. C’est souvent un acte bienveillant et altruiste même si il est extrêmement nuisible à l’image de marque du vendeur du logiciel. Raison pour laquelle les sociétés éditrices incriminées entretiennent volontairement un amalgame entre pirate et hacker.

<http://fr.wikipedia.org/wiki/Hacker>

La consultation de l’encyclopédie libre Wikipedia exige de faire preuve d’esprit critique et de procéder par recoupement de l’information.

- éviter que des personnes mal intentionnées ne prennent connaissance de la faille non corrigée et l’exploitent au préjudice des utilisateurs du logiciel.

Pour ces raisons, un logiciel libre sera toujours plus sûr. Quoi qu’en disent les irréductibles du logiciel propriétaire au sujet des “garanties” offertes par celui-ci.

Ce n’est pas parce qu’un logiciel est libre qu’il est de qualité. Il suffit de parcourir sourceforge.net pour s’apercevoir que le pire côtoie le meilleur. Nous avons vu plus haut que ce qui contribue à la qualité d’un logiciel, c’est la visibilité de son code source et la possibilité de le tester et de l’améliorer. Pour bénéficier de ces améliorations, il faut qu’il soit utilisé par un grand nombre de personnes, certains logiciels restent ignorés, d’autres acquièrent une renommée mondiale. Il est indispensable qu’ils soient entourés du dynamisme des développeurs et de la communauté des utilisateurs.

Les tests logiciels effectués par les équipes de “debuggers” n’ont qu’une valeur symbolique, ils n’offrent aucune certitude. Un logiciel, qu’il soit libre ou propriétaire, est sûr uniquement pour les tests qu’il a subis et seulement dans les conditions dans lesquelles ces tests ont été effectués. C’est à force d’être utilisé, de recevoir de nouveaux rapports de bugs et corrections, qu’un logiciel devient de plus en plus sûr.

Le partage de la connaissance, un des piliers de l’éthique du logiciel libre, permet une évolution plus rapide et moins coûteuse des technologies de l’information. C’est, entre autres, sur cette base qu’il a pris le formidable essor qu’on lui connaît².

Les logiciels libres garantissent quatre libertés fondamentales : la liberté de les utiliser, de les redistribuer, et même de les modifier et de diffuser les versions modifiées. Le citoyen est parfois prompt à protester avec véhémence contre la négation de ses droits, il s’avère être moins vigilant en ce qui concerne les libertés numériques. Il reste un énorme travail de sensibilisation à accomplir, dans le domaine de la vente forcée de logiciels lors de l’achat d’un ordinateur, du respect de la vie privée, de la publication de spécifications à des fins d’interopérabilité et de développement de pilotes, etc.

Comme tous les projets internationaux de développement libre, un logiciel peut subir d’importantes “mutations”. Si on prend l’exemple du plus célèbre des navigateurs libres, Firefox a toute une histoire. Il était au départ la branche libre de Netscape et s’appelait Mozilla, il n’était pas qu’un simple navigateur, c’était aussi :

- un environnement de création de pages html, avec gestion des css, etc ;
- un lecteur de courriel ;
- un carnet d’adresses ;

²Apache représente 80% des serveurs WEB.

– un lecteur de groupes de discussions.

Ce que j'appellerai péjorativement une usine à gaz. Les développeurs l'ont bien compris et ont scindé le navigateur vers une nouvelle branche de développement appelée Phoenix. Phoenix était un navigateur profitant de toutes les fonctionnalités du navigateur Mozilla mais sans sa légendaire lourdeur. Le développement de Phoenix subit une importante mutation en s'orientant vers une autre branche de développement appelée Firebird.

<http://www.mozilla.org/projects/phoenix/phoenix-release-notes.html>

<http://www.mozilla.org/products/firefox/releases/0.7.html>

Suite à la popularité de cette nouvelle branche du navigateur, les fonds provenant de dons, et d'investissements se déplacèrent vers celle-ci. A un point tel que le développement de Mozilla fut abandonné. Toutes les énergies se concentrèrent alors vers une branche commune séparant navigateur, lecteur de courriel et assistant personnel en différents logiciels. C'est la situation actuelle dans laquelle le navigateur s'appelle Firefox.

<http://www.mozilla.org/>

Un logiciel libre peut donc disparaître comme Mozilla, parce que les moyens de son développement ont pris d'autres orientations. Mais c'est pour s'adapter et mieux répondre aux besoins des utilisateurs. Même si il existe quelques rares personnes qui aimaient concevoir leurs pages web (je me demande bien qui ! ;) dans cette usine à gaz qu'était "Mozilla composer", il est nettement préférable que les énergies se concentrent sur un navigateur de qualité.

7.2 Facilité d'installation

Sous GNU/Linux, Koha et PMB posent les mêmes difficultés, qui sont d'installer un système de base et les logiciels et librairies ³ nécessaires à leur fonctionnement.

Le système de base de la plupart des distributions GNU/Linux, s'installe avec aisance, et constituera rarement une difficulté majeure, même pour le néophyte. Il n'en va pas de même pour les logiciels et librairies requises par Koha et PMB, car ces composants ne sont pas tous disponibles en paquets munis de scripts d'installation et de configuration.

C'est la plus grosse difficulté à laquelle sera confrontée la personne désireuse d'installer un de ces SIGB. Il faut réunir et compiler⁴ certains composants logiciels.

La documentation de Koha et PMB décrivant ces opérations est évasive, incomplète, voire partiellement obsolète, en tous cas pour la Debian qui a changé de version depuis plusieurs mois. Ce problème est d'autant plus préoccupant que c'est l'opération la plus difficile à accomplir.

Il m'a dès lors semblé indispensable de consacrer une partie du temps qui m'était imparti à la rédaction de manuels et au développement d'un CDrom automatisant l'installation et la configuration des logiciels et librairies requis par Koha. Je l'ai testé, il est opérationnel,

³Des librairies appelées aussi bibliothèques contiennent des fonctionnalités auxquelles font appel plusieurs programmes. Par exemple, un programme de traitement de texte ne va pas lui-même chercher les caractères dans la mémoire du clavier. Il délègue ce travail à des routines de bas niveau offrant cette fonctionnalité à la plupart des programmes installés sur l'ordinateur qui sont écrits dans le même langage de programmation.

Cf. <http://www.linux-france.org/prj/jargonf/L/libc.html>

⁴Compiler, le mot qui effraie, rien qu'en écrivant ce mot je découragerais probablement une majorité de néophytes.

mais il serait souhaitable qu'il soit testé par des personnes n'ayant aucune connaissance informatique. Faute de temps, je n'ai pas développé une version de CDRom se chargeant de l'installation du système d'exploitation de base, cette tâche reste à accomplir si il s'avère que cette étape pose un problème aux utilisateurs.

Une fois franchie cette étape d'installation des logiciels et librairies dont dépendent ces SIGB, on peut télécharger l'archive du SIGB choisi, la décompresser et exécuter l'assistant l'installation.

Le niveau de difficulté pour installer les logiciels dont dépendent Koha ou PMB est équivalent. Pour Koha, j'ai fait un CDRom d'installation automatisée.

Il suffirait d'en faire un autre pour PMB.

7.2.1 L'assistant d'installation de Koha

La décompression et l'extraction de l'archive de Koha crée un répertoire dans lequel il suffit de se placer pour exécuter le script d'installation.

C'est un script Perl⁵ qui s'exécute en mode texte. Les questions qui y sont posées relèvent à la fois de compétences informatiques et bibliothécaires.

Il est donc souhaitable que les deux compétences soient réunies au moment de cette opération. Si de mauvais choix sont faits, ils sont irréversibles, et il faut recommencer la procédure d'installation.

Par exemple, si on fait le choix du MARC21, il sera impossible de collecter des notices sur un réservoir UNIMARC avec le client z3950.

Du côté des compétences informatiques, il faut juste comprendre la structure d'un chemin UNIX et le principe d'un numéro de port.

Le chemin UNIX se compose d'une racine, représentée par une barre oblique (/) de début. Les autres barres obliques (/) séparent les répertoires et sous répertoires. Le chemin est représenté sous cette forme :

```
/répertoire/sous-répertoire/sous-sous-répertoire/
```

Les ports sont des numéros qui distinguent entre eux les services s'exécutant sur l'ordinateur.

L'usage de cet assistant d'installation ne m'a pas paru évident. Pour aider la personne qui s'en servira, j'ai recréé des copies d'écran de toutes les questions posées. Cette aide se trouve au paragraphe [A.2.2](#).

7.2.2 L'assistant d'installation de PMB

Ici, les choses se passent différemment, l'assistant d'installation, contrairement à l'austère script Perl de Koha, est une jolie interface Web extrêmement conviviale. Chaque étape présente une page Web donnant toutes les explications nécessaires. Au bas de cette page de documentation se trouvent les options d'installation.

⁵Langage de programmation de Koha

Cette convivialité se solde par une difficulté supplémentaire. Contrairement à l'archive de Koha qui peut être extraite n'importe où sur l'ordinateur, l'archive de PMB doit être extraite dans l'arborescence du serveur Web (Cf. `DocumentRoot` au paragraphe B.1.1.1).

L'exécution de l'assistant ne peut se faire qu'après l'avoir appelé via la barre d'adresse d'un navigateur internet. Il faut appeler le script "tables/install.php" ce qui donne comme exemple d'URL :

<http://monserveur.org/tables/install.php>

Cf. <http://www.sigb.net/doc/html-install/ch03s02.html>

La convivialité de l'assistant est telle qu'elle ne requiert quasiment aucune compétence. En contrepartie, l'installation configure PMB pour que l'OPAC s'exécute sur le même port que l'interface bibliothécaire. Par conséquent, on ne peut pas sécuriser PMB en définissant une politique de sécurité différente pour l'OPAC et l'interface bibliothécaire (Cf. paragraphe 3.1.4 au chapitre 3).

7.3 Facilité d'utilisation

7.3.1 PMB

PMB est un logiciel convivial, les fonctionnalités de l'interface bibliothécaire sont aisément accessibles. De plus, elles sont bien documentées.

Quant à l'OPAC, son design est agréable, la possibilité de navigation thématique est une excellente idée.

On peut construire des critères de recherche basés sur des ET, OU, SAUF ; un vrai plaisir.

7.3.2 Koha

L'accès aux fonctionnalités de l'interface bibliothécaire n'est pas toujours aisé. Le design de l'interface est parfois un peu obscur, le menu paramètres est un véritable fourre-tout où certaines fonctionnalités telles que l'importation et l'exportation n'ont pas leur place.

La documentation de l'interface bibliothécaire est en cours de rédaction en anglais.

La recherche avancée dans L'OPAC est moins modulaire que celle de PMB.

Même si la convivialité des interfaces semble être un critère moins important d'un point de vue informatique que la popularité du logiciel, le dynamisme de la communauté des utilisateurs et des développeurs et la qualité du modèle de développement, il serait regrettable que Koha soit rejeté par les utilisateurs à cause de cela.

Si dans le cadre de ce projet il est indispensable de concentrer les efforts sur un seul logiciel, il serait souhaitable d'effectuer des tests auprès des utilisateurs avant son adoption définitive. Décider d'imposer un logiciel (aussi bon soit-il d'un point de vue informatique) dans un tel contexte me semble être un point suffisamment délicat pour prendre toutes les précautions nécessaires afin de s'assurer qu'une quelconque forme de rejet dû au manque de clarté de l'interface bibliothécaire de Koha ne risque pas d'hypothéquer la réussite du projet.

7.4 Robustesse et Sécurité

7.4.1 Koha

La gestion des habilitations des utilisateurs de Koha se base plus sur la confiance que l'on accorde aux bibliothécaires que sur un principe sécuritaire. Un bibliothécaire qui a le droit de modifier les emprunteurs peut s'auto-proclamer administrateur avec tous les droits sur le SIGB. Il faudra veiller à attribuer ce droit uniquement aux bibliothécaires de confiance, méticuleux, et compétents.

Il en va de même pour les outils de configuration où l'on peut aussi bien supprimer une option de configuration que la modifier. Le risque d'endommager ce système par erreur (tag du Marc par exemple) est simplement limité par un avertissement. Il faudra également veiller à attribuer ce droit uniquement aux bibliothécaires de confiance. C'est à cette condition que Koha pourra rester pleinement opérationnel.

Pour les sauvegardes de sécurité, Koha ne propose pas d'outil convivial intégré. Mais ce n'est pas un problème car il existe un autre programme qui permet de le faire (Voir paragraphe 3.1.2.2).

J'ai expliqué au paragraphe 2.3 que le modèle de développement de Koha est évolué et très bien pensé. Le modèle de développement est la méthode de travail employée par les programmeurs pour réaliser l'application. Ce n'est pas parce que la méthode de travail est bonne que le résultat sur le produit fini l'est forcément. Au niveau du résultat j'ai bien constaté quelques petits défauts mineurs que j'ai signalés et qui ont rapidement été corrigés grâce à l'extraordinaire dynamisme de l'équipe de développement de Koha.

7.4.2 PMB

PMB, développé par une société commerciale, accorde visiblement une priorité à la convivialité. L'installateur n'est pas effrayé avec des questions compliquées comme les noms d'hôtes et les ports, mais le résultat final s'en ressent. A moins de modifier la configuration de PMB pour que l'interface bibliothécaire s'exécute sur un port différent de celui de l'OPAC, il sera moins aisé de sécuriser PMB par des politiques de sécurité différentes.

PMB intègre un outil de sauvegarde convivial permettant d'effectuer des backups des tables de la base de données. Il permet de se connecter à un espace de stockage distant par FTP (protocole de transfert de fichiers entre ordinateurs en réseau). J'ai trouvé cet outil extrêmement sympathique même si son intégration à un SIGB n'est pas vraiment indispensable dans la mesure où il existe d'autres outils conviviaux s'acquittant parfaitement de cette tâche (Voir paragraphe 3.1.2.2).

7.5 Import/export des données

Les échanges de données entre les SIGB se font dans un format de fichier standardisé qu'est l'ISO2709. Koha et PMB incluent une fonction d'importation/exportation de notices. Ces notices sont stockées dans un fichier au format ISO2709. L'importation peut aussi se faire à partir d'un réservoir auquel on accède grâce à un client Z3950.

Les outils d'importation de Koha et PMB sont tout à fait satisfaisants et jouent parfaitement leur rôle pour autant que la source de données respecte les normes en vigueur (ISO2709).

Les données provenant de CDS/ISIS posent par contre quelques problèmes. De nombreuses tentatives de récupération de ces données ont été effectuées, elles n'ont pas encore abouti à des résultats satisfaisants.

Ce problème est en partie du au format du fichier d'exportation. Ce problème pourrait se résoudre mais faute de temps toutes les voies n'ont pas été explorées. Mais il faut ajouter que cette absence de solution satisfaisante n'est pas un critère disqualifiant pour Koha, mais plutôt pour le logiciel CDS/ISIS.

7.6 Les aides

7.6.1 Installation

En ce qui concerne l'installation sous GNU/Linux, les aides sont tout aussi insatisfaisantes pour Koha que pour PMB. Comme nous l'avons vu au paragraphe 7.2 la tâche la moins aisée à accomplir est l'installation des logiciels et librairies dont dépendent les SIGB. C'est malheureusement un sujet mal documenté. La documentation de PMB décrit par exemple la compilation des libraires Yaz dont dépend le client Z3950. Si il y a bien une tâche à laquelle le néophyte ne désire pas s'atteler c'est la compilation, et à juste titre d'ailleurs. Car dans le cas où elle pose le moindre problème (ce qui arrive souvent quand il manque des librairies), une personne non initiée sera incapable de comprendre les erreurs et de les résoudre. Ces opérations de compilation sont trop compliquées (ou en tous cas le deviennent au moindre problème), même si elles étaient parfaitement décrites (ce qui n'est même pas le cas).

La seule solution acceptable, c'est de faire en sorte que l'installateur ne doive pas les accomplir, raison pour laquelle j'ai fait un cdrom qui se charge de cette partie du travail pour l'installation de Koha.

7.6.2 Utilisation

Le manuel de l'utilisateur de Koha est en cours de rédaction⁶ Vous n'y trouverez que le premier chapitre. Quant aux manuels de l'utilisateur et de l'administrateur de PMB, il est tout à fait satisfaisant⁷.

En contrepartie de son absence de documentation exhaustive, il existe des listes de diffusion sur Koha en français et en anglais. Elles sont d'un dynamisme remarquable et permettent de résoudre la plupart des problèmes rencontrés.

7.7 Pérennité et possibilité d'évolution

Nous avons vu au paragraphe 7.1 que le dynamisme des développeurs et de la communauté des utilisateurs dépend de la popularité d'un logiciel. Koha et PMB sont bien sûr soumis à

⁶<http://www.kohadocs.org/usersguide/>

⁷<http://www.sigb.net/doc/html-user/index.html>
<http://www.sigb.net/doc/html-admin/index.html>

cette règle selon leurs propres spécificités.

Koha est un logiciel libre traduit en plusieurs langues, sa communauté de développeurs est mondiale.

Conséquences :

Visibilité, popularité et dynamisme planétaire, Koha a toutes les chances d'évoluer et sa pérennité est assurée.

C'est une auberge espagnole à l'échelle planétaire, chacun peut y apporter une amélioration et la proposer à l'équipe de développement.

PMB est un logiciel libre, développé par une société commerciale française dont l'interface est traduite en anglais mais sa base de développement est française.

Conséquences :

Dans les codes sources de PMB, certaines variables et noms de fonctions sont nommés en français, d'autres en anglais. Un programmeur qui ne connaît pas le français ne va rien comprendre au code source. Mais on peut les traduire assez facilement avec des recherches et remplacements, pour améliorer la lisibilité du code source.

PMB sera développé en fonction de décisions d'intérêt commercial prises par la société. Par exemple, selon le rapport de l'ENSSIB, Eric Robert a expliqué qu'à ce jour, SARL PMB Service ne souhaite pas travailler pour des bibliothèques universitaires.

L'exemple de Mozilla démontre qu'il est possible qu'un logiciel comme Koha subisse d'importantes "mutations" et entraîne dans sa disparition des fonctionnalités délaissées car faisant double emploi avec d'autres logiciels spécialisés et mieux adaptés. Mais ce sera pour prendre des orientations correspondant mieux aux évolutions des besoins de la communauté d'utilisateurs.

7.8 Conclusion

Pour rappel, PMB a été écarté dans l'étude menée par l'ENSSIB pour les raisons suivantes :

1. il a exclusivement été développé pour des petites bibliothèques insérées dans un réseau BDP.
2. Eric Robert a expliqué qu'à ce jour SARL PMB Service ne souhaite pas travailler pour des bibliothèques universitaires.
3. il ne permet pas de cataloguer en respectant les différents formats MARC
4. il permet difficilement l'indexation à partir d'un répertoire structuré de notices autorisés
5. PMB n'est actuellement soutenu que par une société commerciale.

Malgré les quatre premiers critères d'exclusion cités, PMB est utilisé avec succès à l'université de Lubumbashi depuis novembre 2004. Alors que le succès de l'usage de Koha reste à démontrer.

Quant au dernier critère, rien n'interdit à la société commerciale de cesser de développer ou de décider de développer les futures versions sous licence propriétaire.

Quoi qu'il en soit, toutes les versions existant sous une licence libre le resteront définitivement. L'importante communauté d'utilisateurs continuera donc à exister. Rien n'empêche celle-ci d'assurer le développement libre de nouvelles version de PMB.

D'un point de vue strictement informatique, la méthode de travail employée par les programmeurs pour réaliser Koha est très évoluée. Elle me semble supérieure à celle utilisée pour développer PMB, bien que faute de temps et d'une documentation satisfaisante sur le sujet, je n'ai pas eu l'occasion d'étudier en détail le modèle de développement de PMB ni d'en faire une étude technique détaillée. Toutefois, l'interface bibliothécaire de PMB est plus conviviale et accessible que celle de Koha qui souffre d'un certain manque de clarté, de convivialité et de structure.

Si les quatre premiers critères d'exclusion vous semblent incontournables le choix de Koha s'impose (en tenant compte des remarques faites au paragraphe 7.3.2 à propos du manque de convivialité de l'interface bibliothécaire de Koha).

Bien que mon CDRom facilite considérablement l'installation de Koha, il n'inclut pas l'installation automatique du système d'exploitation. Cette tâche reste à réaliser. Voir paragraphe 7.2 page 64.

Au cas où l'installation automatisée de Koha échouerait, il faudrait prévoir un minimum de 3 jours de support technique et de formation par site pour effectuer une installation manuelle.

Si les quatre premiers critères d'exclusion vous semblent discutables, l'usage de PMB pourrait être envisagé sous certaines conditions :

- je n'ai pas mené d'étude technique approfondie de PMB, cette tâche serait à accomplir.
- il serait nécessaire de développer un CDRom qui automatise l'installation des logiciels et librairies dont dépend PMB. L'installation manuelle est trop compliquée sous Debian GNU/Linux car elle exige un travail de compilation.

Comme pour Koha, au cas où l'installation automatique de PMB (avec un CDRom) échouerait, il faudrait prévoir un minimum 3 jours de support technique et de formation par site.

Annexe A

Description de l'installation de Koha

Ce chapitre décrit l'installation de Koha qu'il est possible de réaliser de deux manières différentes :

1. **En utilisant mon cdrom d'installation.**

Ce cdrom rend les services suivants :

- (a) Il évite de devoir télécharger Koha ;
- (b) Il met à votre disposition et installe tous les logiciels et librairies dont Koha a besoin pour pouvoir fonctionner.
- (c) Il lance le script d'installation de Koha. Si vous utilisez mon cdrom, vous aurez 3 opérations à accomplir :

- i. Installer le système de base ;
- ii. télécharger¹ et graver l'ISO du cdrom².

Une image ISO est un fichier que l'on grave sur un support amovible comme un cdrom ou un dvdrom. On peut aussi l'écrire sur une clé USB. Une fois ce fichier est gravé sur le support et que celui-ci est placé dans le lecteur, on a accès à l'arborescence de l'ensemble des répertoires, sous répertoires et fichiers du cdrom.

- iii. placer le cdrom dans le lecteur et le monter. Sous DOS ou Windows le montage des périphérique est toujours transparent. Quand vous allez sur un lecteur de cdrom, le système d'exploitation attache le système de fichier du cdrom à celui du système de manière transparente. Sous GNU/Linux le même principe s'applique mais vous pouvez aussi le faire manuellement avec la commande mount. C'est ce que vous serez contraint de faire dans ce cas, car vous partez d'un système de base réduit aux fonctions essentielles.

```
sh# mount /media/cdrom
sh# cd /media/cdrom/
sh# sh installkoha.sh
```

A. La première commande monte le cdrom ;

B. la deuxième se déplace sur le cdrom

¹L'adresse de téléchargement n'est pas encore déterminée, elle vous sera communiquée par courriel lors de l'envoi de ce document.

²Il n'est pas obligatoire de graver l'ISO du cdrom vous pouvez en faire l'économie en montant l'image. Cf. paragraphe [E.2.1](#) en annexe [E](#)

C. le troisième lance le script d'installation automatisée des composants nécessaires au fonctionnement de Koha et koha lui même.

iv. Il vous suffit ensuite de répondre aux questions posées par l'assistant d'installation de Koha. Cet assistant est un script fourni avec Koha qui automatise son installation. Cette tâche doit être effectuée par un informaticien en présence du bibliothécaire responsable de la migration vers le SIGB Koha. A ce stade passez directement au paragraphe [A.2.2](#).

2. En suivant pas à pas les opérations suivantes afin d'effectuer une installation manuelle.

A.1 Télécharger koha

Tout d'abord, il faut choisir la version de Koha que l'on désire installer, ce choix peut s'effectuer sur la page d'accueil du projet Koha sur sourceforge.net.

Pour trouver l'adresse du projet Koha on peut soit utiliser le moteur de recherche de sourceforge.net soit aller sur <http://www.koha.org/> et suivre le lien "download Koha", ce qui nous conduit à cette adresse :

http://sourceforge.net/project/showfiles.php?group_id=16466

Ensuite, il faut choisir la version de Koha, à l'heure où j'écris ce document il s'agit de la 2.2.3, mais cela aura peut-être changé demain !

En suivant le lien de la version de son choix (ex Download koha-2.2.3.tar.gz), sourceforge.net propose de choisir un miroir local. Un miroir est un serveur secondaire qui permet d'alléger la charge du serveur principal, pour une zone géographique donnée. Par exemple le serveur le plus proche de la Belgique est celui d'OVH en France. Si vous ne trouvez pas de miroir correspondant à votre zone géographique ou même à votre continent, choisissez en un à votre convenance. Dans ce cas précis l'URL est le suivant :

<http://ovh.dl.sourceforge.net/sourceforge/koha/koha-2.2.3.tar.gz>

Télécharger l'archive soit :

– Avec la commande `wget`.

A partir de la machine cliente disposant d'une interface graphique et d'un navigateur comme Firefox, il suffit de repérer l'URL de l'archive à télécharger comme expliqué ci dessus.

Ensuite dans une console ssh (Cf. paragraphe [D.2.2.1](#)), on colle le lien par un clic du milieu après la commande `wget`. Cette opération ne requiert pas les droits d'administrateur.

```
sh$ cd /tmp
```

```
sh$ wget http://switch.dl.sourceforge.net/sourceforge/koha/koha-2.2.3.tar.gz
```

Ici par exemple j'ai choisi le serveur de Lausanne en Suisse.

– Avec un navigateur en mode texte

```
sh$ w3m koha.org
```

Avec les flèches on déplace le curseur sur [home-blobby-down1] ensuite sur go to Sourceforge to download Koha. choisir la version de Koha, le serveur miroir et télécharger.

– En copiant sur le serveur une archive préalablement téléchargée sur la machine cliente :

```
sh$ scp /chemain/koha-2.2.2b.tar.gz root@139.165.158.59:/opt/
```

A.2 Installer Koha

Se placer dans le répertoire d'extraction et décompresser l'archive.

- l'archive a été déposée dans /tmp
 - elle sera décompressée dans /opt
 - si /opt n'existe pas on le crée :

```
sh# mkdir /opt
sh# cd /opt
tar -xzvf /tmp/koha-2.2.2b.tar.gz
```
- Voir le manuel de tar.

A.2.1 Exécuter le script d'installation

Se placer dans le répertoire de l'archive de koha :

```
sh# cd /opt/koha-2.2.2b
```

Exécuter le script d'installation : `sh# ./installer.pl`

Le point qui précède la barre oblique désigne le chemin absolu du répertoire dans lequel on se trouve. Puisqu'on se trouve dans "/opt/koha-2.2.2b" le point "." remplace "/opt/koha-2.2.2b". En fait le point veut dire "ici". Ceci est nécessaire pour exécuter les programmes dont le chemin d'exécution n'est pas défini par la variable \$PATH

A.2.2 L'assistant d'installation de Koha

Le script d'installation interactif pose quelques questions. La plupart de options par défaut conviennent. Il faut donc simplement les valider.

J'ai reproduit dans ce paragraphe les copies d'écran de l'entièreté de la procédure d'installation de Koha. Si aucune instruction n'est donnée, c'est que l'option proposée par défaut convient. Dans ce cas, il suffit d'appuyer sur la touche Entrée du clavier pour valider.

WELCOME TO THE KOHA INSTALLER

Are you ready to begin the installation? ([Y]/N):

PERL MODULES

Checking perl modules ...

PERL MODULES AVAILABLE

All required perl modules are installed.

Press <ENTER> to continue:

OPAC DIRECTORY

OPAC Directory [/usr/local/koha/opac]:

LIBRARIAN DIRECTORY

Intranet Directory [/usr/local/koha/intranet]:

LOG DIRECTORY

Le répertoire des log sera installé dans `/var/log` pour mieux respecter les standards.

Koha Log Directory [`/usr/local/koha/log`]: `/var/log/koha`

Lorsque la valeur par défaut convient, validez simplement.

DATABASE NAME

Nom de la base de données mysql qui sera utilisée par Koha.

Database name [Koha]:

DATABASE HOST

Donnez le nom du serveur de base de données si il est installé sur une autre machine :

Database host [localhost]:

DATABASE USER

Le login utilisé par le lecteur/administrateur par défaut de l'interface web de Koha. Bien sûr il sera possible d'en créer d'autres avec l'interface d'administration.

Database user [kohaadmin]:

DATABASE PASSWORD

Ici le script d'installation demande un mot de passe pour ce login.

Choisissez un mot de passe sûr et facile à retenir. Voir astuce au paragraphe [E.2.2](#) de l'annexe E.

Ce mot de passe vous sera demandé pour vous connecter à l'interface bibliothécaire de Koha.

Password for database user kohaadmin: votremotdepasse

APACHE CONFIGURATION

Les informations qui seront demandées dans cette partie concernent la configuration du serveur web apache. Ces informations seront utilisées par le script d'installation pour créer le fichier `/etc/koha-httpd.conf` qui contient les directives d'apache nécessaires au fonctionnement de Koha.

WEB E-MAIL CONTACT

Adresse de courrier électronique de l'administrateur du serveur :

E-mail contact [webmaster@]: votreadresse@domaine.com

WEB HOST NAME OR IP ADDRESS

L'adresse IP ou le nom d'hôte de la machine.

Ici, il est possible de créer des hôtes virtuels, c'est à dire utiliser des noms d'hôtes différents de celui du serveur. Malheureusement, le script d'installation ne pose qu'une seule fois cette question et le nom d'hôte virtuel est, dans ce cas, le même pour l'OPAC et l'interface d'administration, seuls les ports diffèrent. Si l'on veut que le nom d'hôte de l'interface d'ad-

ministration soit différent de celui de l'OPAC, il faut se référer à la section [B.1.1](#) en Annexe. Si vous désirez utiliser un nom d'hôte plutôt qu'une IP (Vous trouverez des explications sur les notions d'IP et de domaine au paragraphe [3.1.3.3](#)) l'administrateur du réseau dans lequel se trouve la machine devra le créer. En admettant qu'il ai créé un nom d'hôte tel que "koha-admin.organisation.org" :

```
Host name or IP Address [dg11]: koha-admin.organisation.org
```

Avant ou après avoir exécuté le script d'installation de Koha, il est nécessaire de modifier le fichier de configuration d'apache. Pour les détails, se référer à la section [B.1.2](#) en Annexe.

OPAC PORT

Le port sur lequel va s'exécuter un service est comparable aux numéros de maison dans une rue. Il existe une convention qui lie des numéros de port à certains services. Cette convention indique par exemple que le port 80 est réservé au service web "http". Pour obtenir une liste des services et le port qui y est associé, on peut consulter le fichier `/etc/services` avec la commande :

```
sh$ less /etc/services
```

Par défaut le port 80 convient très bien.

```
Enter the OPAC Port [80]:
```

LIBRARIAN PORT

Le script d'installation propose par défaut d'utiliser le port 8080 pour l'intranet. En consultant `/etc/services` on s'aperçoit que le port 8080 est par convention affecté à un autre service :

```
webcache          8080/tcp          # WWW caching service
```

Il ne serait donc pas judicieux de l'utiliser pour Koha. On lui affectera le port 8082.

```
Enter the Intranet Port [8080]: 8082
```

UPDATING APACHE CONFIGURATION

Le script va maintenant procéder à l'installation de Koha, voici un extrait des messages qui s'affichent à l'écran :

```
Checking for modules that need to be loaded...
```

INSTALLING FILES

```
Copying files to installation directories:Copying admin templates to /usr/local/koha
Copying admin interface to /usr/local/koha/intranet/cgi-bin.
Copying main scripts to /usr/local/koha/intranet/scripts.
Copying perl modules to /usr/local/koha/intranet/modules.
Copying OPAC templates to /usr/local/koha/opac/htdocs.
Copying OPAC interface to /usr/local/koha/opac/cgi-bin.
```

MYSQL ROOT USER PASSWORD

Pour pouvoir créer la base de données, le script d'installation a besoin du mot de passe de l'utilisateur root de Mysql³ qui est le login administrateur (Cf. paragraphe D.2.4) :

password:

CREATING DATABASE

Creating the MySQL database for Koha...

MARC FIELD DEFINITIONS

Le choix du format MARC doit être défini une fois pour toutes lors de l'installation. Si vous comptez utiliser un serveur Z39.50 afin d'y collecter des notices, vous ne pourrez accéder qu'aux réservoirs correspondant au format que vous avez choisi.

Si vous avez l'intention d'accéder à des réservoirs francophones comme celui de la BNF, il est préférable de faire l'installation en UNIMARC [2].

You can import MARC settings for:

1 MARC21

2 UNIMARC

N none

NOTE: If you choose N, nothing will be added, and you must create them all yourself. Only choose N if you want to use a MARC format not listed here, such as DANMARC. We would like to hear from you if you do.

*** UPGRADE *** If you UPGRADE your version from a previous 2.x.x, the right choice here is N (None) to preserve your local MARC setup.

Choose MARC definition [1]: 2

ADD BRANCH AND PRINTER

Would you like to describe an initial branch and printer? [Y]/N:

Branch Name [MAIN]:

Branch Code (4 letters or numbers) [MAIN]:

Printer Name [lp]:

Printer Queue [/dev/lp0]:

CHOOSE LANGUAGE

This version of koha supports a few languages.

³Ne pas confondre avec le login unix root

en : default language, all pages available

fr : complete translation

es : partial librarian site translation

zh_TW : partial translation

en is used when a screen is not available in your language

If you specify a language here, you can still change it from the system preferences screen in the librarian sit.

Which language do you choose? fr

SELECT SQL DIRECTORY

Select a directory. You will see every file included in this directory and be able to
This is a VERY important feature. By selecting the proper options, you can get a pre
Choose wisely.

0 => FINISHED

1 => english

2 => french

2

SELECT SQL FILE

Select a file to append to the Koha DB.

enter a number. A detailed explanation of the file will be given
if you confirm, the file will be added to the DB

0 => FINISHED

1 => autorites_norme_unimarc.txt

2 => framework_ANCI.txt

3 => framework_BASE.txt

4 => framework_CART.txt

5 => framework_DEFAULT.txt

6 => framework_ELEC.txt

7 => framework_MICR.txt

8 => framework_MONO.txt

9 => framework_PERI.txt

10 => framework_PROJ.txt

11 => framework_TIRE.txt

12 => stopwords.txt

13 => unimarc_standard_systemprefs.txt

13

Ici il est préférable de prendre tout par précaution, pour cela, entrez 13, puis l'assistant d'installation vous demandera :

INSERT FRENCH/UNIMARC_STANDARD_SYSTEMPREFS.TXT ?

This file contains systemprefs for french UNIMARC standard setup

Confirm loading of this file into Koha [Y]/N:

Validez simplement et la liste des fichiers à ajouter à la base de données vous sera à nouveau présentée. Choisissez 12 puis validez, revenez à la liste. L'assistant vous demande à nouveau de sélectionner un répertoire de fichiers SQL à inclure dans la base de données de Koha. Et ainsi de suite jusqu'à 0 pour terminer.

SELECT SQL DIRECTORY

Select a directory. You will see every file included in this directory and be able to
This is a VERY important feature. By selecting the proper options, you can get a preview
Choose wisely.

0 => FINISHED

1 => english

2 => french

1

On peut maintenant quitter la boucle de cette partie de l'installation des fichiers SQL à ajouter à la base de données de Koha en entrant 0.

RESTART APACHE

The web server daemon needs to be restarted to load the new configuration for Koha. The installer can do this if you are using Apache and give the root password.

Would you like to try to restart Apache now? [Y]/N:

Ici on valide simplement.

AUTHENTICATION

This release of Koha has a new authentication module. You will be required to log in to access some features.

IMPORTANT: You can log in using the userid and password from the /etc/koha.conf configuration file at any time. Use the "Members" screen to add passwords for other accounts and set their flags.

Press the <ENTER> key to continue:

Comme indiqué on valide simplement.

INSTALLATION COMPLETE

Congratulations ... your Koha installation is complete!

You will be able to connect to your Librarian interface at:

`http://dg10.lib.ulg.ac.be:8082/`

use the koha admin mysql login and password to connect to this interface.
and the OPAC interface at:

`http://dg10.lib.ulg.ac.be:80/`

NOTE: You need to add lines to your main httpd.conf to include
/etc/koha-httpd.conf and to make sure it is listening on the right ports
(using the Listen directive). Then, restart Apache.

Please read the Hints file and visit <http://www.koha.org> (in english) or
www.koha-fr.org (in french) Press <ENTER> to exit the installer:

Ici on peut s'assurer que l'on a sélectionné les bonnes options.

En ce qui concerne la note sur "httpd.conf" Le script qui automatise l'installation des programmes (voir paragraphe 1.1.4.1) nécessaires à Koha a déjà effectué cette configuration.

Pour finir l'installation il ne faut pas oublier de suivre les instructions détaillées en annexe au paragraphe B.1.2.

Annexe B

Le serveur Web Apache

B.1 Configuration du serveur Apache

B.1.1 Les hôtes virtuels

Le script d'installation de Koha ne propose pas de configurer des hôtes virtuels différents pour l'OPAC et l'interface d'administration. Il est possible de le faire en éditant le fichier `/etc/koha-httpd.conf`.

Partons du principe que nous disposons de deux noms de domaine pointant vers l'adresse IP du serveur :

- `koha.lib.ulg.ac.be` pour L'OPAC sur le port 80
- `koha-admin.lib.ulg.ac.be` pour l'interface d'administration sur le port 8082

Le nom d'hôte du serveur sur lequel on installe Koha :

`dg10.lib.ulg.ac.be`

Son IP

`139.165.158.46`

Ajouter les directives demandant à Apache d'écouter sur les port 80 et 8082

```
Listen 80
```

```
Listen 8082
```

Décommander la ligne :

```
NameVirtualHost *
```

Mettre une étoile à la place de l'adresse IP permet de ne rien modifier dans ce fichier de configuration au cas où le serveur changerait d'IP. Normalement on indique à la suite de cette directive l'adresse IP qui correspond au nom d'hôte du serveur, ou plus exactement celle des interfaces par lesquelles s'effectueront les connexions http.

Voici un exemple de configuration pour l'hôte virtuel de l'OPAC utilisant les informations qui précèdent :

```
<VirtualHost *:80>
```

```
    ServerAdmin bbarbier@ulg.ac.be
```

```

DocumentRoot /usr/local/koha/opac/htdocs
ServerName koha.lib.ulg.ac.be
ScriptAlias /cgi-bin/koha/ /usr/local/koha/opac/cgi-bin/
Redirect permanent index.html http://koha.lib.ulg.ac.be:80/cgi-bin/koha/opac-main
ErrorLog /var/log/koha/opac-error_log
TransferLog /var/log/koha/opac-access_log
SetEnv PERL5LIB "/usr/local/koha/intranet/modules"
SetEnv KOHA_CONF "/etc/koha.conf"
</VirtualHost>

```

B.1.1.1 Détail des directives

```
<VirtualHost adresse[:port] ...> ... </VirtualHost>
```

Utilisé pour “encapsuler” un groupe de directives qui s’appliquent à un hôte virtuel particulier.

ServerAdmin Définit l’adresse e-mail que le serveur inclut dans tout message d’erreur retourné au client.

DocumentRoot

Définit le répertoire racine à partir duquel httpd va distribuer les fichiers.

ServerName

C’est ici que sera indiqué l’hôte virtuel. Cette directive impose que le nom d’hôte apparaisse dans l’en-tête “Host :” d’une requête pour être associé à cet hôte virtuel. Ou plus simplement, l’URL utilisé par le navigateur pour accéder à L’OPAC.

ScriptAlias

Cette directive a le même comportement que la directive “Alias” mais elle diffère de celle-ci par le fait que le contenu du répertoire vers lequel pointe l’alias doit obligatoirement être interprété comme du script (Perl, PHP, Python, etc).

L’utilité de cette directive :

La directive “DocumentRoot” de l’OPAC indique le chemin à partir duquel Apache doit chercher ses fichiers “Web”.

Une autre directive présente par défaut dans la configuration d’Apache (DirectoryIndex), indique que si rien d’autre que le nom d’hôte ou l’IP de la machine n’est indiqué, Apache doit rechercher un des fichiers : index.html, index.htm, index.shtml, index.cgi, ou index.php dans le répertoire désigné par la directive “DocumentRoot”.

Un fichier nommé “index.html” se trouve bien dans le répertoire “/usr/local/koha/opac/htdocs” il affiche simplement “WELCOME TO KOHA” et il contient l’instruction :

```
<meta http-equiv="refresh" content="0; url=/cgi-bin/koha/opac-main.pl">
```

Cette instruction va “re-diriger” la navigation vers “/cgi-bin/koha/opac-main.pl”.

Dans ce cas, la directive “ScriptAlias”, va mettre en correspondance l’URL “/cgi-bin/koha/”¹ avec le répertoire “/usr/local/koha/opac/cgi-bin/” du système de fichier.

Si une requête telle que :

http://139.165.158.59/cgi-bin/koha/opac-main.pl est émise, Apache(Perl en

¹Ce chemin n’existe nulle part sur le système de fichier

réalité) va interpréter le fichier “opac-main.pl” se trouvant dans le répertoire “/usr/local/koha/opac/cgi-bin”.

Redirect
(A FAIRE)

Cette directive ne fonctionne pas dans les hôtes virtuels créés par le script d’installation de Koha.

ErrorLog

Fichier dans lequel vont être écrits les logs d’erreur d’Apache pour l’hôte virtuel de l’OPAC. Par défaut sur une Debian, Apache écrit ses logs d’erreur dans le fichier /var/log/apache/error.log

TransferLog

Fichier dans lequel vont être écrits les logs d’accès d’Apache pour l’hôte virtuel de l’OPAC. Par défaut sur une Debian, Apache écrit ses logs d’accès dans le fichier /var/log/apache/access.log

SetEnv

Directive qui passe des variables d’environnement aux scripts CGI. C’est pour cette raison qu’Apache a besoin du module mod_env (Cf. paragraphe B.1.2).

B.1.2 Modification du fichier httpd.conf

Le script d’installation ne modifie pas le fichier de configuration d’Apache :

/etc/apache/httpd.conf, il faut y inclure /etc/koha-httpd.conf par la directive “Include” pour que ce qui s’y trouve soit pris en compte.

Il est donc nécessaire d’éditer le fichier /etc/apache/httpd.conf et lui ajouter la ligne :

```
Include /etc/koha-httpd.conf
```

Ce n’est pas tout, il faut aussi charger le module “mod_env.so” qui est utilisé par Apache pour passer des variables d’environnement (telles que celles contenues dans /etc/koha-httpd.conf) aux scripts CGI². On charge ce module avec la directive “LoadModule” en ajoutant cette ligne :

```
LoadModule env_module /usr/lib/apache/1.3/mod_env.so
```

Pour autant que le module se trouve bien dans le répertoire : /usr/lib/apache/1.3/.

Pour le trouver :

- Mettre à jour la base de données de l’ensemble des noms des fichier existant sur le système :

```
sh# updatedb
```

- Faire une recherche dans cette base de données : sh# locate mod_env.so

```
/usr/lib/apache/1.3/mod_env.so
```

Après avoir modifié la configuration du serveur :

- vérifier la syntaxe de son fichier de configuration (optionnel) :

```
sh# apachectl configtest
```

- le redémarrer :

```
sh# /etc/init.d/apache restart
```

²Koha est écrit en Perl CGI

Annexe C

Le système de gestion de base de données MySQL

C.1 Sauvegardes des données

C.1.1 Copie avec tar

Par défaut (selon la hiérarchie des arborescences sous Debian) les répertoires de bases de données sont localisés dans `/var/lib/mysql/`.

`/var/lib/mysql/` contient un répertoire par base de données.

Les fichiers de la base de données Koha (Koha étant le nom de la base de données par défaut proposé lors de l'installation) se trouveront donc dans : `/var/lib/mysql/Koha`.

```
sh$ su -
sh# cd /var/lib/mysql/Koha
sh# tar -cvf /var/backups/backupKoha_db-17-05-2005.tar *
```

Pour de plus amples informations sur les commandes GNU/Linux `su`, `tar` et `bzip2` on peut consulter leur manuel en tapant dans un terminal les commandes :

```
sh$ man su
sh$ man tar
sh$ man bzip2
```

C.1.2 mysqlhotcopy

`mysqlhotcopy` est script PERL qui utilise l'API Perl-DBI.

Cf. <http://dbi.perl.org/>

Voici un exemple d'utilisation de la commande `mysqlhotcopy` :

```
mysqlhotcopy -u root --password=mot_de_passe Koha /tmp/
```

Cette opération va recopier tout les fichiers présents dans le répertoire `/var/lib/mysql/Koha` vers le répertoire `/tmp/Koha`. Ces fichiers seront utilisables tels quels par le serveur à partir du

nouveau répertoire /tmp/Koha. <http://dev.mysql.com/doc/mysql/fr/mysqlhotcopy.html>

C.1.3 mysqldump

La méthode de sauvegarde des données la plus courante, est d'utiliser le script mysqldump, utilitaire qui permet d'exporter une base ou un groupe de bases vers un fichier texte :

<http://dev.mysql.com/doc/mysql/fr/mysqlhotcopy.html>

Cette commande ne s'utilise pas dans une console SQL mais dans un shell UNIX.

Exemple :

```
sh$ mysqldump [options] db_name [tables]
```

Si vous ne spécifiez pas de table, la base de données complète sera exportée.

Exemple :

```
sh$ mysqldump -c -u root -pmotdepasse Koha --tables borrowers > backup-file.sql
```

Les options les plus courantes sont :

-single-transaction

Cette option ajoute la commande SQL BEGIN avant d'exporter les données. Ce mode va exporter l'état de la base au moment de la commande BEGIN sans bloquer les autres applications.

-complete-insert, -c

Crée un fichier de sauvegarde sous la forme de commandes SQL d'insertion, plus lent lors de la restauration des données, mais la syntaxe sql le rend compatible avec la plupart des SGDB¹.

Exemple :

```
INSERT INTO marc_subfield_structure
(tagfield, tagsubfield, liblibrarian, libopac, repeatable, mandatory,
kohafield, tab, authorised_value, authtypecode, value_builder, seealso,
frameworkcode, isurl, hidden)
VALUES ('010', 'a', 'LC control number', 'LC control number', 0, 0,
'biblioitems.lccn', 0, '', '', '', '', '', NULL, NULL);
```

-user=nom_utilisateur, -u nom_utilisateur

-password[=password], -p[password]

Ne pas laisser d'espace entre l'option et le mot de passe.

Nom de la base de données. Ici Koha

-tables

Tout ce qui suit cette option est considéré comme des noms de tables, -tables doit donc se trouver en dernier dans la ligne de commande.

Voici quelques autres options couramment employées.

¹ Système de Gestion de Base de Données

```
-no-create-info, -t
```

Ne recrée pas les tables avec la commande SQL CREATE TABLE en vue de la restauration des données, dans des tables existantes.

```
-no-data, -d
```

Ne récupère pas les données mais uniquement la structure des tables.

Pour importer les données dans la base MySQL :

```
sh$ mysql basededonnee < backup-file.sql
```

Mais on peut aussi compresser une archive :

```
bzip2 -9 mysqldump.dump
```

On peut bien sûr la restaurer sans que l'archive décompressée ne crée un fichier sur le disque dur. Le gain en place et en temps est énorme.

```
sh$ bzcat mysqldump.dump.bz2 | mysql -u root -p motdepasse test
```

Pour cela, au lieu d'envoyer le flux de décompression du fichier vers la sortie standard comme le fait bzip2, on tube vers MySQL qui l'intègre directement dans la base de données.

Si l'on désire éditer le fichier d'archive comprimé avant de le restaurer dans MySQL il faut alors procéder en deux étapes :

Décompresser le fichier :

```
sh$ bunzip2 mysqldump.dump.bz2
```

Et ensuite le restaurer à l'aide d'une redirection "<".

```
mysql -u root -p motdepasse test < mysqldump.dump
```

Pour de plus amples informations sur la redirection et les pipelines consultez le manuel de bash.

```
sh$ man bash
```

Ensuite recherchez les paragraphes : "Pipelines" et "REDIRECTION".

Pour faire une recherche dans le man taper "/MotClé" puis validez. Pour passer à l'occurrence suivante appuyez sur "n" comme next.

C.1.4 SELECT

Pour faire des sauvegardes sélectives en envoyant le résultat d'une requête dans un fichier :

```
sql> SELECT * INTO OUTFILE 'nom_fichiers'  
-> FROM nom_de_table where condition;
```

Pour assurer la cohérence des données à l'instant de la requête, il faut verrouiller la table que l'on souhaite sauvegarder.

Voir la documentation de mysql sur les verrous :

<http://dev.mysql.com/doc/mysql/fr/lock-tables.html>

Voici un exemple de requête sur un lecteur/administrateur de la table des utilisateurs en posant un verrou sur celle-ci :


```

sql> FLUSH TABLES borrowers;
sql> LOCK TABLES borrowers READ;
sql> SELECT * INTO OUTFILE '/tmp/testbebackup.bkp'
-> FROM borrowers where userid='barbier';
sql> UNLOCK TABLES;

```

1. La première instruction purge du cache les tables données en argument(ici borrowers).
2. Un verrou en lecture (READ) permet aux autre threads de continuer à effectuer des requêtes.
3. Requête sql dont le résultat est redirigé dans le fichier '/tmp/testbebackup.bkp'.
4. Déverrouillage de la table

Pour restaurer les archives dans une autre base :

```

sql> USE autre_base;
sql> BEGIN;
sql> LOAD DATA INFILE '/tmp/testbebackup.bkp' INTO TABLE borrowers;
sql> COMMIT;

```

La commande USE autre_base connecte la console à une autre base de données(autre_base).

La commande BEGIN va initialiser une transaction. Si une erreur est survenue lors de la restauration des données, on peut revenir à l'état qui précède le début de la transaction avec la commande ROLLBACK.

C.1.5 Mise en oeuvre d'une réplication des serveurs

Il faut bien sûr disposer de deux ordinateurs, le plus puissant servant de serveur maître. Celui-ci entretient des log binaires de toutes les modifications qui surviennent. L'esclave met à jour ses bases de données d'après les log binaires créés par le maître.

Pour que la réplication soit une solution intéressante, il est indispensable d'installer une version de mysql 4.0 ou supérieure.

En effet, dans les versions antérieures, la fonction LOAD DATA FROM MASTER ne peut pas fonctionner sans bloquer le maître, car elle exige de le verrouiller durant son appel pour assurer la cohérence des données.

C.1.5.1 Règles et erreurs à ne pas commettre lors de la réplication avec MySQL

1. Si l'on désire ne répliquer qu'une seule base, binlog-do-db=UneSeuleBase ; **On ne peut pas indiquer plusieurs bases en argument de cette option** sans quoi aucun évènement ne s'inscrira dans le log binaire, et donc aucune réplication n'aura lieu, excepté avec la commande SQL :

```

sql> LOAD DATA FROM MASTER

```

 Cette commande n'utilise pas de log binaire.²
2. Après l'installation du serveur maître, vous trouverez dans le fichier "/etc/mysql/my.cnf" une option qui empêche le serveur d'accepter une connexion autre que locale. Pour mettre en oeuvre la réplication il est indispensable que le serveur esclave puisse se connecter au maître. Il faut commenter l'option :

```

#bind-address = 127.0.0.1

```

 En ajoutant un dièse (#) au début de la ligne.

²J'ai perdu de nombreuses heures à comprendre ce qui n'allait pas avec cette option.

Cette information se trouve dans le fichier :
 “/usr/share/doc/mysql-server-4.1/README.Debian.gz”.

* NETWORKING:
 =====

For security reasons, the Debian package has enabled networking only on the loopback device using "bind-address" in /etc/mysql/my.cnf. Check with "netstat -tlnp" where it is listening. If your connection is aborted immediately see if "mysqld: all" or similar is in /etc/hosts.allow and read hosts_access(5).

3. Sur l’esclave, le serveur ne tient pas compte des options de réplication inscrites dans le fichier “/etc/mysql/my.cnf” si il existe un fichier “master.info”. Ce fichier a priorité aussi bien sur le fichier de configuration que sur les options de la ligne de commande. Si vous utilisez la commande `sql CHANGE MASTER TO` pour changer des options de réplication, elles seront inscrites dans “master.info”. Pour que MySQL puisse lire sa configuration dans “/etc/mysql/my.cnf”, il faut supprimer le fichier “master.info” qui se trouve généralement dans le répertoire des données, sous Debian :

```
sh# rm /var/lib/mysql/master.info
```

4. Si vous relancez le serveur maître, il change de fichier log, il faudra s’assurer que l’esclave à été informé de ce changement.

C.1.5.2 Opération à accomplir sur le serveur maître :

1. Créer un utilisateur MySQL spécial pour la réplication sur le maître.
2. Configurer le fichier my.cnf
3. Faire une sauvegarde des données du maître

C.1.5.2.1 Création de l’utilisateur de réplication : Se connecter en tant qu’administrateur à la base de données et taper ceci dans la console SQL :

```
sql> GRANT REPLICATION SLAVE, RELOAD, SUPER, SELECT ON *.*  
TO repl@'%' IDENTIFIED BY 'motdepasse';
```

Cet utilisateur doit posséder les droits de :

REPLICATION SLAVE

RELOAD

SUPER

SELECT

On attribue ces droits sur toutes les bases dans ce cas(`ON *.*`)³ à l’utilisateur repl qui pourra se connecter depuis n’importe quel autre serveur (`@'%'`).

³On pourrait les restreindre à la base Koha (`ON koha.*`), ou à certaines tables de la base (`koha.table`)

C.1.5.2.2 Configurer le fichier my.cnf du maître Pour que l'esclave puisse se connecter au serveur maître il faut éditer le fichier /etc/mysql/my.cnf et mettre en commentaire l'option `bind-address` en mettant un dièse “#” au début de la ligne :

```
#bind-address          = 127.0.0.1
```

À la section `[mysqld]`, ajoutez les options `log-bin`, `server-id` (serveur-id doit recevoir un entier positif unique).

```
server-id              = 1
log-bin                = /var/log/mysql/mysql-bin.log
#max_binlog_size      = 104857600
binlog-do-db           = Koha
```

Dans le cadre de cette analyse on ne réplique que la base Koha avec l'option `binlog-do-db`, pour limiter la taille maximum du log binaire on peut enlever le dièse de commentaire devant l'option `max_binlog_size`.

Il est très important que les identifiants “server-id” soient tous différents. Pensez à server-id comme à une valeur comparable à une adresse IP : elle identifie de manière unique un serveur dans la communauté des réplicateurs.

On peut aussi demander à MySQL de produire des log texte afin d'observer le bon déroulement des opérations :

```
log                    = /var/log/mysql.log
```

N'oubliez pas de mettre cette ligne en commentaire lorsque le serveur sera en production, car ces log nuisent aux performances.

Redémarrez le serveur.

C.1.5.2.3 Faire une sauvegarde des données du maître Il existe plusieurs solutions pour effectuer cette sauvegarde, la plus simple, dans ce cas, est une copie de tous les fichiers de la base que l'on veut répliquer en utilisant `mysqlhotcopy`, (cette méthode est décrite au paragraphe C.1.2) ou effectuer l'opération manuellement avec `tar` (Cf. paragraphe C.1.1), voir la documentation de MySQL.

<http://dev.mysql.com/doc/mysql/fr/replication-howto.html>

On peut aussi utiliser `mysqldump` (voir paragraphe C.1.3), c'est la méthode la plus simple.

```
sh$ mysqldump -u root --master-data -p motdepasse Koha > /tmp/mysql-snapshot.dump
```

Copier l'archive dans le répertoire /tmp/ de l'esclave :

```
sh# scp /tmp/mysql-snapshot.dump root@ordi.esclave.net:/tmp/
```

C.1.5.3 Opération à accomplir sur le serveur esclave :

1. Configurer le fichier my.cnf de l'esclave.
2. Réinitialiser le fichier de configuration master.info
3. Créer la base Koha et la mettre à jour.

4. Lancer les threads esclaves.

C.1.5.3.1 Configurer le fichier my.cnf Éditer le fichier /etc/mysql/my.cnf de l'esclave :

1. Enlever le commentaire “#” de la ligne suivante :

```
log                = /var/log/mysql.log
```

MySQL écrira des log au format texte dans /var/log/mysql.log pour surveiller son comportement lors de la configuration, lorsque le système est en production, il est préférable de remettre le “#” du commentaire.

Attention sur une Debian les log relatifs aux échecs de connexion seront écrits dans “/var/log/syslog”. On peut les surveiller avec la commande :

```
sh# tail -f /var/log/syslog
```

Ou “-f” comme follow, de cette façon l'écran se rafraîchit pour afficher les nouvelles lignes de log.

2. Donner un identifiant unique à chaque serveur :

```
server-id          = 2
```

Le maître a reçu l'id 1, celui de l'esclave doit être différent.

3. Spécifier l'hôte ou l'IP du maître de réplication :

```
master-host=maitre.de.replication.com
```

4. Indiquer le nom d'utilisateur et mot de passe que l'esclave utilise lors de l'identification auprès du maître (Cf. paragraphe C.1.5.2.1.) :

```
master-user=repl
```

```
master-password=motdepasse
```

5. Le nombre de secondes qu'un esclave attend avant de tenter de se reconnecter au maître, dans le cas où le maître et l'esclave perdent la connexion.

```
master-connect-retry=60
```

6. Le nom à utiliser pour le fichier dans lequel l'esclave stocke les informations sur le maître.

```
master-info-file=/var/lib/mysql/master.info
```

7. Le nom d'hôte ou l'adresse IP de l'esclave, qui doit être indiqué lors de l'enregistrement de l'esclave chez le maître.

```
report-host=dg10.lib.ulg.ac.be
```

Relancer le serveur :

```
sh# /etc/init.d/mysql restart
```

C.1.5.3.2 Réinitialiser le fichier de configuration master.info Se connecter en tant qu'administrateur à la base de données de l'esclave en entrant la commande :

```
sql> STOP SLAVE;
```

```
sql> RESET SLAVE;
```

C.1.5.3.3 Créer la base Koha et la mettre à jour Voici les commandes à entrer dans un shell, pour mettre la base de données à jour on utilise l'archive que l'on avait créée sur le maître et copiée sur l'esclave.

```
sh$ mysqladmin -u root -p create Koha
```

```
sh$ mysql -u root -p Koha < /tmp/mysql-snapshot.dump
```

C.1.5.3.4 Lancer les threads esclaves. `sql> START SLAVE;`

Annexe D

Debian GNU/Linux

D.1 Installation de la Debian Sarge

D.2 Installation des logiciels de base

Au lieu d'effectuer le travail rébarbatif qui suit, je vous propose d'utiliser un script. Je laisse les explications pour que vous puissiez répondre aux questions posées par celui-ci.

D.2.1 Liste des logiciels à installer

Voici les logiciels de base à installer :

- gnupg : logiciel de cryptographie et signature à clefs asymétrique
- locales : paquet permettant de supporter les différentes langues
- ntpdate : logiciel qui se connecte à un serveur de temps pour synchroniser l'horloge de l'ordinateur
- gcc : compilateur C
- ssh : voir paragraphe [D.2.2.1](#)
- unzip : décompression des archives .zip
- make : permet de structurer les instructions de compilation
- lynx : navigateur web en mode texte
- w3m : idem
- wget : outil de téléchargement et de création de site miroir
- ncftp : client ftp
- ftplib : idem
- less : visionneuse de texte ascii
- mysql-server-4.1 : serveur de base de données
- apache : serveur web

Perl et ses bibliothèques :

- perl
- libdbi-perl, libdbd-mysql-perl, libdate-manip-perl, libhtml-template-perl, libmail-sendmail-perl
- libxml-sax-perl

Des bibliothèques nécessaires au fonctionnement du client z3950, la gestion du XML, la cryptographie :

- libyaz-dev yaz-doc libxml2-dev libssl-dev libwrap0-dev
- libevent-perl libmd5-perl libpdf-api2-perl

D.2.2 Prérequis

Cette étape implique que vous ayez déjà une installation minimale d'un système GNU/Linux. Il est recommandé d'utiliser un noyau de la branche 2.4 ou supérieur sans quoi vous aurez quelques problèmes avec le module PERL Net : :Z3950.

Si vous installez une Debian Sarge au prompt du CD d'installation `boot` : tapez `bf24` puis appuyez sur "enter".

D.2.2.1 Méthode de connexion

Si vous êtes à proximité du serveur sur lequel vous désirez installer koha vous pouvez utiliser une connexion locale.

Dans le cas contraire, il vous faudra l'effectuer via une connexion distante. Pour ce faire, il faut que la machine cible (celle sur laquelle on veut installer Koha) dispose d'un serveur ssh, pour l'installer il faut taper dans un terminal¹ en root :

```
sh# apt-get install ssh
```

Pour obtenir d'avantage d'informations sur cette commande on peut consulter son manuel en tapant dans un terminal :

```
sh$ man apt-get
```

Le protocole ssh permet d'établir une connexion distante sécurisée entre deux ordinateurs. L'échange des mots de passe est crypté ainsi que tous les transferts. Lorsqu'une connexion ssh est établie, le client se retrouve dans un terminal ouvert sur la machine cible, exactement comme si il utilisait le clavier de cette machine².

D.2.2.1.1 Le système d'exploitation de la machine à partir de laquelle on effectue la connexion :

- Si l'installation se fait au départ d'un ordinateur distant tournant sous GNU/Linux, il suffit d'utiliser un client ssh.

Par exemple si la machine sur laquelle on veut installer Koha a l'IP 139.165.158.59

```
ssh root@139.165.158.59
```

- Si l'installation se fait à partir d'un ordinateur distant tournant sous MS Windows, il faut utiliser PuTTY.

<http://www.chiark.greenend.org.uk/%7Esgtatham/putty/>

La liste des logiciels à installer se trouve au paragraphe [D.2.1](#)

Cette installation se fait en utilisant la commande `apt-get` :

Dans le terminal (shell) il faut taper :

```
sh# apt-get install gnupg locales ntpdate gcc ssh w3m
```

```
sh# apt-get install unzip make lynx wget ncftp ftp less
```

¹sh# n'étant pas le début de la commande mais la représentation du prompt de notre shell voir chapitre [iii](#)

²Il peut même y faire de l'export display, et disposer de tous les logiciels graphiques installés sur la machine cible.

Il est aussi utile d'installer un éditeur de texte qui servira lors de la modification des fichiers de configuration : `sh# apt-get install vim-common`

D.2.3 Installation de Perl

Perl (Practical Extraction and Report Language) est un langage de programmation dérivé des scripts shell, créé en 1986 par Larry Wall afin de mettre au point un système de News entre deux réseaux. Il s'agit d'un langage interprété dont l'avantage principal est d'être très adapté à la manipulation de chaînes de caractères. De plus, ses fonctionnalités de manipulation de fichiers, de répertoires et de bases de données en ont fait le langage de prédilection pour l'écriture d'interfaces CGI.

Koha a été développé en Perl pour fonctionner sur un serveur L.A.M.P (Linux Apache Mysql Perl/Python/PHP), il dépend de l'interpréteur et de quelques bibliothèques Perl. Certaines d'entre elles sont disponibles en paquet Debian, (on peut les installer en utilisant la commande `apt-get`), d'autres devront être installées avec l'utilitaire "Perl CPAN".

D.2.3.1 Installer Perl et les bibliothèques disponibles en paquet Debian :

```
sh# apt-get install perl
sh# apt-get install libdbi-perl libdbd-mysql-perl libdate-manip-perl
libhtml-template-perl libmail-sendmail-perl
```

D.2.3.2 Installer les autres bibliothèques avec l'utilitaire "Perl CPAN" :

Lors de son premier usage (dans ce cas-ci pour installer la bibliothèque "MARC : :Record") l'utilitaire "Perl CPAN" pose quelques questions, généralement l'option par défaut convient, ce n'est donc pas très compliqué.

Il faut taper la commande suivante dans un terminal :

```
sh# perl -MCPAN -e 'install "MARC::Record"'
```

Ici, il suffit bien sûr de valider.

```
Are you ready for manual configuration? [yes]
```

Ici aussi l'option par défaut peut convenir.

```
CPAN build and cache directory? [/root/.cpan]
```

J'ai augmenté la valeur mais rien ne dit que la valeur par défaut soit insuffisante.

```
Cache size for build directory (in MB)? [10] 50
```

On laisse l'option par défaut pour les questions suivantes.

```
Perform cache scanning (atstart or never)? [atstart]
```

```
Cache metadata (yes/no)? [yes]
```

```
Your terminal expects ISO-8859-1 (yes/no)? [yes]
```

```
File to save your history? [/root/.cpan/histfile]
```


Number of lines to save? [100]

Si un module CPAN détecte une dépendance, il peut la compiler automatiquement “follow” ou bien demander une confirmation “ask” et enfin l’ignorer (ne choisissez “ignore” que si vous savez exactement ce que vous faites).

Policy on building prerequisites (follow, ask or ignore)? [ask]

Ici aussi, j’ai pris le temps de vérifier plusieurs fois le chemin de chaque binaire, sur une Debian, il correspond bien à ce qui est indiqué³.

On valide donc sans soucis.

```
Where is your gzip program? [/bin/gzip]
Where is your tar program? [/bin/tar]
Where is your unzip program? [/usr/bin/unzip]
Where is your make program? [/usr/bin/make]
Where is your lynx program? [/usr/bin/lynx]
Where is your wget program? [/usr/bin/wget]
Where is your ncftpget program? [/usr/bin/ncftpget]
Where is your ftp program? [/usr/bin/ftp]
Where is your gpg program? [/usr/bin/gpg]
What is your favorite pager program? [/usr/bin/less]
What is your favorite shell? [/bin/bash]
```

Pour les questions suivantes on laisse les valeurs par défaut(à moins de disposer de serveur multi-processeur).

```
Your choice: [INSTALLDIRS=site]
Parameters for the 'make' command?
Typical frequently used setting:

    -j3                dual processor system
Your choice: []
Your choice: []
Timeout for inactivity during Makefile.PL? [0]
```

A moins d’accéder à internet via un proxy(ce qui fort peut probable puisque dans ce cas, on est sur un serveur), on ignore les questions suivantes.

```
Your ftp_proxy?
Your http_proxy?
Your no_proxy?
You have no /usr/local/lib/perl/cpan/sources/MIRRORED.BY
    I'm trying to fetch one
LWP not available
CPAN: Net::FTP loaded ok
Fetching with Net::FTP:
    ftp://ftp.perl.org/pub/CPAN/MIRRORED.BY
```

³En réalité lynx pointe vers “/etc/alternatives/lynx” qui est structure de sélection de binaire propre à Debian, mais on n’a pas à s’en préoccuper

Choix du miroir CPAN :

Cette analyse à été réalisée en Belgique, dans ce cas, le choix sera :

```
4 pour le continent
2 pour le pays
4 3 2 1 5 pour le miroire
pas d'autre URL
```

- (1) Africa
- (2) Asia
- (3) Central America
- (4) Europe
- (5) North America
- (6) Oceania
- (7) South America

Select your continent (or several nearby continents) [] 4

- (1) Austria
- (2) Belgium
- (3) Bosnia and Herzegovina
- (4) Bulgaria
- (5) Croatia
- (6) Czech Republic
- (7) Denmark
- (8) Estonia
- (9) Finland
- (10) France
- (11) Germany
- (12) Greece
- (13) Hungary
- (14) Iceland
- (15) Ireland
- (16) Italy

16 more items, hit SPACE RETURN to show them

Select your country (or several nearby countries) [] 2

- (1) ftp://cpan.mirrors.skynet.be/pub/CPAN
- (2) ftp://ftp.belnet.be/packages/cpan/
- (3) ftp://ftp.easynet.be/pub/CPAN/
- (4) ftp://ftp.kulnet.kuleuven.ac.be/pub/mirror/CPAN/
- (5) ftp://ftp.scarlet.be/pub/cpan/

Select as many URLs as you like (by number),

put them on one line, separated by blanks, e.g. '1 4 5' [] 4 3 2 1 5

Enter another URL or RETURN to quit: []

L'utilitaire affichera les messages de téléchargement et de compilation et si tout se passe bien terminera par ceci :

```
Writing /usr/local/lib/perl/5.8.4/auto/MARC/Record/.packlist
Appending installation info to /usr/local/lib/perl/5.8.4/perllocal.pod
/usr/bin/make install -- OK
```

Il se pourrait qu'une dépendance ne soit pas satisfaite.

Voici un cas de figure où la bibliothèque Zlib n'a pas été trouvée :

```
-----
Unsatisfied dependencies detected during [A/AR/AREIBENS/PDF-API2-0.44.tar.gz]
-----
Compress::Zlib
Shall I follow them and prepend them to the queue
of modules we are processing right now? [yes]
```

Si vous avez choisi l'option "ask", il suffit de valider en appuyant sur la touche entrée.

D.2.3.3 La boîte à outils Yaz

Yaz est une boîte à outils⁴ qui implémente le protocole Z39.50.

Yaz existe en paquet Debian, mais pas dans la distribution officielle, elle nécessite une modification du fichier de configuration : "/etc/apt/sources.list".

C'est dans ce fichier que sont répertoriés les miroirs du système de paquetage Debian. Ce système de paquetage rend accessible l'ensemble des binaires, librairies, scripts et tout ce qui peut être installé via la commande `apt-get`

Pour pouvoir installer la boîte à outils Yaz avec la commande `apt-get` il faut :

1. Ajouter 3 lignes à la fin du fichier "/etc/apt/sources.list", en utilisant l'éditeur nano, qui a l'avantage de présenter en bas de l'écran les commandes possibles en fonction du contexte, il est donc le plus convivial des éditeurs sous Linux.

```
sh# nano /etc/apt/sources.list
ajouter :

# for Yaz Toolkit
deb http://www.indexdata.dk/debian indexdata/woody released
deb-src http://www.indexdata.dk/debian indexdata/woody released
```

Ctrl-O pour enregistrer, valider et Ctrl-X pour quitter

2. Pour que le système de paquetage Debian prenne en compte les modifications :

```
sh# apt-get update
```

3. Installation des paquets :

```
sh# apt-get install libyaz-dev yaz-doc
```

Le système propose de satisfaire les dépendances de ces paquets, il suffit de valider.

Les NOUVEAUX paquets suivants seront installés :

```
libssl-dev libssl0.9.6 libwrap0-dev libxml2 libxml2-dev libyaz
libyaz-dev yaz-doc
```

0 mis à jour, 8 nouvellement installés, 0 à enlever et 0 non mis à jour.

Il est nécessaire de prendre 6737ko dans les archives.

Après dépaquetage, 18,7Mo d'espace disque supplémentaires seront utilisés.

```
Souhaitez-vous continuer ? [O/n]
```

⁴traduction de "toolkit" qui désigne dans ce contexte, des librairies de développement

D.2.3.4 Terminer l'installation des modules Perl

```
sh# perl -MCPAN -e 'install "Event"'
sh# perl -MCPAN -e 'install "Net::Z3950"'
sh# perl -MCPAN -e 'install "Digest::MD5"'
```

Ces commandes afficheront beaucoup de messages à l'écran, si il y a des dépendances à satisfaire, il suffit de valider.

Exemple :

```
Warning: prerequisite Compress::Zlib 1 not found.
Writing Makefile for PDF::API2
This is PDF::API2, Version 0.41 (2005-03-25 14:29:04)
---- Unsatisfied dependencies detected during
[A/AR/AREIBENS/PDF-API2-0.41.tar.gz] -----
    Compress::Zlib
Shall I follow them and prepend them to the queue
of modules we are processing right now? [yes]
```

Si tout s'est bien passé :

```
Installing /usr/local/man/man3/PDF::API2::Basic::TTF::Head.3pm
Installing /usr/local/man/man3/PDF::API2::Version.3pm
Installing /usr/local/man/man3/PDF::API2::Basic::TTF::AATKern.3pm
Writing /usr/local/lib/perl/5.8.4/auto/PDF/API2/.packlist
Appending installation info to /usr/local/lib/perl/5.8.4/perllocal.pod
/usr/bin/make install -- OK
```

D.2.4 Installation et configuration de MySQL

MySQL est le SGBD utilisé par Koha.

<http://dev.mysql.com>

C'est un système de gestion de base de données plus simple que Postgresql mais plus rapide.

Il est disponible en paquet Debian en deux versions :

mysql-server qui est la version par défaut en Debian testing⁵.

mysql-server-4.1 C'est la version de Mysql qui est utilisée dans le cadre de cette analyse, bien que la version antérieure soit pleinement compatible avec Koha, la version antérieure ne correspond pas toujours à la documentation en ligne.

<http://dev.mysql.com/doc/mysql/fr/index.html>

Certaines commandes comme DROP USER n'existent qu'à partir de cette version.

<http://dev.mysql.com/doc/mysql/fr/removing-users.html>

C'est aussi la version recommandée sur la page de téléchargement.

Pour installer Mysql :

```
sh# apt-get install mysql-server-4.1
```

Comme souvent, il y a des dépendances à satisfaire, on valide simplement la proposition :

⁵Pour rappel ce document se base sur une Debian testing et pas sur la Woody bien qu'elle soit considérée comme la version stable

Les NOUVEAUX paquets suivants seront installés :

```
libdbd-mysql-perl libdbi-perl libmysqlclient12 libnet-daemon-perl
libplrpc-perl mysql-client mysql-server
```

0 mis à jour, 7 nouvellement installés, 0 à enlever et 0 non mis à jour.

Il est nécessaire de prendre 5175ko dans les archives.

Après dépaquetage, 12,7Mo d'espace disque supplémentaires seront utilisés.

Souhaitez-vous continuer ? [O/n]

Le script d'installation du paquet montrera un écran d'information.

Il faut attribuer un mot de passe au compte administrateur de Mysql, ce compte se nomme "root", à ne pas confondre avec le compte administrateur homonyme du système d'exploitation.

```
sh mysqladmin password lemotdepasse
```

Ce mot de passe sera demandé par le script d'installation de Koha

D.2.5 Installation et configuration d'Apache

Apache est le serveur Web le plus utilisé, il est maintenu par l'Apache Software Foundation qui regroupe plusieurs dizaines de projets Open source de grande ampleur, tel que Ant, Jakarta et le célèbre framework Struts pour le développement d'applications en java.

Pour l'installer :

```
sh# apt-get install apache
```

D.3 Installation du système de base

D.4 La gestion différée des tâches

D.4.1 Le gestionnaire at

At est un gestionnaire occasionnel, c'est à dire qu'on doit indiquer la date et l'heure à laquelle la tâche doit être lancée une seule fois, ce qui le différencie de cron qui est conçu pour lancer des tâches récurrentes (toutes les minutes, toutes les heures, tous les jours, tous les mois).

Il y a trois solutions pour lancer une tâche différée avec at :

- en ligne de commande et une commande :

```
sh$ at 21:30 29.12.2000 < /home/benoit/scripts/win
```

- en précisant un fichier où se trouvent les commandes avec l'option -f

```
sh$ at -f /tmp/test_at 21:45 29.12.2000
```

- en utilisant l'éditeur de at qui n'est vraiment pas convivial

```
sh$ at 21:52 29.12.2000
```

```
at> echo "Il est 21:52 message de at et son éditeur infecte" >/dev/tty0
at> <EOT>
```

```
warning: commands will be executed using /bin/sh
job 106 at 2000-12-29 21:52
```

Attention, ne pas oublier d'indiquer le chemin complet vers le script ou la commande à exécuter.

D.4.2 Le gestionnaire cron

Cron est un gestionnaire de tâche récurrentes, il se gère avec la commande `crontab`

Attention sous X11(le protocole X11 gère l'affichage graphique sur les systèmes UNIX), il faut toujours préciser le nom de l'utilisateur.

Pour ouvrir la crontab de root(administrateur du système)

```
sh# crontab -u root -e
```

Cette crontab est un simple fichier texte dans lequel on ajoute les commandes à exécuter à une heure donnée. Bien sûr la rédaction de ce fichier se conforme à une syntaxe particulière. Exemple, pour lancer chaque nuit à 23h30 un script "backupkoha" chargé d'effectuer une sauvegarde de la base de données Koha :

```
30 23 * * * /root/scripts/backupkoha
```

Attention, ne pas oublier d'indiquer le chemin complet vers le script ou la commande à exécuter.

-u précise le nom de l'utilisateur

-e (edit user's crontab)

-l (list user's crontab)

-r (delete user's crontab)

Lors de l'appel de la commande `crontab`, il y aura ouverture de l'éditeur de texte `vim` si il est installé (voir paragraphe D.5).

D.4.2.1 description détaillée de la syntaxe d'une crontab

- Chaque utilisateur du système dispose de sa crontab
- La commande ainsi que ses paramètres est écrit sur une ligne
- les 5 premiers champs servent à positionner l'exécution de la commande dans le temps, ils peuvent être remplacé par une étoile.
- chaque ligne est divisée en 6 champs dont voici la description :

1. Minute[0-59] de l'heure à laquelle on veut exécuter la commande.
Toutes les minutes si remplacé par une étoile.
2. Heure[0-23] à laquelle on veut exécuter la commande.
Toutes les heures si remplacé par une étoile.
3. jour du mois [1-31] auquel on veut exécuter la commande.
Tous les jours si remplacé par une étoile.
4. mois de l'année [1-12] auquel on veut exécuter la commande.
Tous les mois si remplacé par une étoile.

5. jour de la semaine [1-7] auquel on veut exécuter la commande.

Tous les jours si remplacé par une étoile.

6. commande à exécuter, précédée du chemin complet dans l'arborescence des répertoires.

– si l'on veut énumérer plusieurs valeurs de temps, elles doivent être séparées par une virgule, ou indiquée par intervalle séparé d'un tiret "-".

– chacun de ces champs doit être séparé par un espace ou une tabulation.

La disposition se présente comme suit :

```
* * * * * /root/scripts/commande
min heure date mois jour_de_la_semaine commande à exécuter
```

D.4.2.2 Exemples :

Pour exécuter une commande toutes les minutes :

```
* * * * * /root/scripts/commande
```

une commande exécutée à 15,16 et 17 h tous les jours ouvrables

```
* 15,16,17 * * 1-5 /root/scripts/commande
```

Pour rappel, si on veut indiquer plusieurs valeur par champs, il faut les séparer par une virgule ou spécifier l'intervalle séparée par un tiret.

D.4.3 Références

Pour de plus amples informations sur l'utilisation de cron, consultez les manuel avec les commandes suivantes :

```
sh$ man cron
sh$ man 5 crontab
sh$ man 1 crontab
```

D.5 L'éditeur de texte vim

Pour éditer le fichier toto tapez cette commande :

```
sh# vim toto Ouvre le fichier toto
sh# vim -R toto Ouvre en lecture seule
```

Une fois dans l'éditeur vim, il faut comprendre qu'il existe plusieurs modes de fonctionnement :

- Le mode Commande, dans lequel vous vous trouvez quand vous ouvrez vim. Dans ce mode, vous tapez des commandes... que nous verrons plus loin ! Si vous êtes dans un autre mode et que vous voulez revenir au mode commande, tapez Echap.
- Le mode Insertion auquel on accède par la touche Inser. L'indicateur – INSERT – apparaît alors en bas de l'écran. Dans ce mode, vous insérez du texte classiquement.
- Le mode Remplacement auquel on accède en appuyant une deuxième fois sur Inser. L'indicateur – REPLACE – apparaît alors en bas de l'écran. Dans ce mode, le texte entré remplace le texte présent sous le curseur.

– Le mode Visuel auquel on accède par la touche `v` depuis le mode Commande. L'indicateur – VISUAL – apparaît alors en bas de l'écran. Ce mode permet de sélectionner du texte pour y appliquer globalement des commandes.

Voici une liste des commandes les plus utilisées. Il faut bien entendu être en mode Commande pour les taper :

```
:new Ouvre un nouveau buffer vierge.
:w toto Enregistrer sous toto
:e toto Ouvre le fichier toto
:r toto Insert le fichier toto dans le texte en cours

/chaîne Recherche d'une chaîne vers le bas
?chaîne Recherche d'une chaîne vers le Haut
n Répète la dernière commande de recherche
N Répète la commande de recherche vers le haut
n Répète la dernière commande de remplacement
% Recherche la parenthèse fermante (pour c et lisp)
:g/toto[/p] Liste les lignes contenant toto dans le fichier
:2,9g/toto[/p] Liste les lignes de 2 à 9 contenant toto

:3,8s/old/nouv/g Substitution de tous les old par nouv de la ligne 3 à 8
:g/old/s//nouv/g Substitution de tous les old par nouv du fichier
:g/old/s//nouv/gp Idem mais avec la liste des lignes modifiées
dd Pour supprimer une ligne
p Colle le contenu du buffer (sous la ligne courante)
u Undo
Ctrl-r Redo
10yy Copie 10 lignes (p pour paste)
10dd Coupe 10 lignes (p pour paste)

:set ai Indentation auto
:set noai Sans indentation auto
```


Annexe E

Annexes diverses

Sont rassemblées ici les parties de textes qui ne trouvent leur place ni en annexes thématiques, ni dans le texte lui-même.

E.1 Définitions

E.1.1 Image ISO

Une image ISO est un fichier que l'on grave sur un support amovible comme un cdrom ou un dvdrom. Une fois ce fichier gravé sur le support et celui-ci placé dans le lecteur, on a accès à l'arborescence de l'ensemble des répertoires, sous répertoires et fichiers du cd/dvdrom.

E.1.2 Miroir

Un miroir est un serveur secondaire qui permet d'alléger la charge du serveur principal, pour une zone géographique donnée. Bien sûr rien ne vous empêche de télécharger sur le serveur principal si vous craignez que le miroir ne soit pas à jour.

E.1.3 URL

Un URL est un format de nommage universel pour désigner une ressource sur Internet.(Uniform Resource Locator)

Exemple :

<http://koha-admin.lib.ulg.ac.be:8082/cgi-bin/koha/mainpage.pl>

Dans ce cas, l'URL est composé de 4 parties :

Protocole	Hôte	Port	Chemin sur le serveur
http ://	koha-admin.lib.ulg.ac.be	:8082	/cgi-bin/koha/mainpage.pl

- La première partie indique à quel type de service s'adresse la requête, ici le web.
- La deuxième à quel ordinateur sur internet s'adresser, on aurait pu utiliser son IP. Cette partie est le nom d'hôte complet aussi appelé FQDN pour Fully Qualified Domain Name. C'est un nom complet du serveur jusqu'au domaine, en passant par les sous-domaines, il est en effet plus facile de se souvenir de "koha-admin.lib.ulg.ac.be" que de l'IP 139.165.158.46.

- La troisième indique le port, par défaut une requête http(web) émanant d'un client passe par le port 80. Mais on peut en préciser un autre. Ainsi on dispose d'un port pour l'OPAC sur le 80 d'un autre pour l'intranet sur le port 8082.
- Et enfin le chemin vers le fichier sur le serveur.

Bien que dans ce cas il s'agisse d'une URL absolue, le chemin sur le système de fichiers du serveur commence à partir de celui indiqué dans la directive DocumentRoot. C'est un chemin relatif au système de fichier local au serveur et une URL absolue du point de vue du client web (Voir au paragraphe B.1.1 en Annexe)

E.1.4 Règles de normalisation

Sont des règles à respecter lors de la conception de bases de données.

Très schématiquement et sans tenir compte de leurs subdivisions, les plus importantes de ces règles disent ceci :

- Valeur atomique des attributs
Tout attribut contient une valeur atomique. Cela signifie que les attributs à valeur multiple sont interdits.
Par exemple une entité personne ne peut contenir le nom de ses enfants sous la forme :
Prénom_enfants : Ludivine, Étienne, Cédric.
En effet on ne peut mettre qu'une seule valeur par occurrence de ce champ.
- Identifiant unique
Tout individu doit obligatoirement posséder un identifiant (voir aussi la définition d'une clé primaire au paragraphe E.1.7).
- Les propriétés doivent avoir un sens pour chacune des occurrences de l'entité.
Par exemple une étagère et une chaise sont des meubles que l'on pourrait stocker dans le même ensemble. Si on en fait une table, elle ne pourrait pas avoir un attribut nombre d'étagères, car il n'aurait pas de sens pour une chaise.

E.1.5 Le modèle relationnel

Dans un SGBD¹ informatisé, une table est comparable à la description d'une fiche d'un fichier.

Prenons par exemple la description de la table "personnes" caractérisée par un nom, un prénom et un identifiant unique :

PERSONNES
Id_pres :
Nom :
Prenom :

Chaque fiche est une occurrence de cette table.

Les tables sont représentées par les SGBD sous l'aspect d'une grille contenant des lignes et des colonnes.

¹ Système de Gestion de Base de Données

Le titre des colonnes, appelé attribut, correspond aux champs de la table ou aux libellés des champs d'une fiche d'un fichier.

Les lignes appelées tuples (parfois appelées objets ou enregistrement) correspondent aux occurrences de la table.

Ses occurrences sont représentées comme ceci :

PERSONNES		
Id_pers	Prénom	Nom
0	Linus	Torvalds
1	Richard	Stallman
2	Eric	Raymond

La plupart des informations stockées dans la base de données de Koha le sont sous cette forme, c'est à dire que des tables spécifiques contiennent des données correspondant au même domaine.

Les emprunteurs sont stockés dans la table borrowers, le prénom est encodé dans le champs prédéfini, il en va de même pour le nom.

E.1.6 Méta données

Une métadonnée est une donnée servant à définir ou décrire une autre donnée. Le format MARC par exemple utilise des informations standardisées pour décrire les données bibliographiques.

Les notices sont utilisées pour décrire le contenu d'un document. Elles facilitent la gestion interne des ressources documentaires et, côté usagers(OPAC), permettent d'optimiser la recherche et la localisation des documents.

Prenons comme exemple d'information standardisées le Tag "100" et son sous champs "a" du MARC21, il désigne le nom de l'auteur. "100" "a" "Personal name" sont des données enregistrées dans la base de données (et pas comme un champ de la table) qui disent que la valeur "Erich Gamma" est un nom d'auteur. Ce sont des données qui décrivent d'autres données.

E.1.6.1 Exemple d'application simplifié

Prenons par exemple le cas d'un magasin de meubles qui voudrait encoder la liste, la nature, et la description des produits qu'il vend.

Il y a des armoires qui doivent être décrites par :

- l'aspect des matériaux utilisé (bois, verre, etc) ;
- le nombre d'étagères ;
- la finition (verni, ou peinture) ;
- la couleur ;
- la hauteur ;
- la profondeur ;

– la largeur ;

Il y a des coussins qui doivent être décrits par :

– son rembourrage (kapok, coton, mousse, etc) ;

– le tissu ;

– le motif du tissu ;

– la couleur ;

Il y a des lampes qui doivent être décrites par :

– les dimensions

– la puissance maximum

– nombre d’ampoules

– type de socket.

On pourrait faire une gigantesque table produits avec toutes les propriétés nécessaires pour décrire tous les articles du magasin... Toutefois, une des règles fondamentale des bases de données relationnelles dit : **Les propriétés doivent avoir un sens pour chacune des occurrences de l’entité.**

Donc ça ne va pas, le nombre d’ampoules et le type de socket n’a pas de sens pour une occurrence de coussin, pas plus que le rembourrage pour une occurrence de lampe.

Il serait tout aussi insensé de vouloir créer une table séparée pour chaque produit vendu avec des propriétés propres à celui-ci, et rappeler le programmeur pour ajouter une table dans la base de données chaque fois qu’un nouveau produit est disponible.

On va donc avoir recours aux **méta données**.

E.1.6.2 Principe relationnel des méta données

Pour décrire un coussin, ou un meuble on ne va pas utiliser des libellés de champs prédéfinis, comme on vient de le voir au paragraphe précédent, mais on va se ménager la possibilité de les encoder dans un champ nommé par exemple “libellé”.

MEUBLES			
Id	Noms	LibelléDescrip	ValDescrip
0	Coussin	Rembourrage	Kapok
1	Étagère	matériau	bois

Dans cet exemple on encode le nom du libellé de la description comme une valeur et non plus comme un champ prédéfini.

*Bien sûr ce n’est qu’une version simplifiée, pour la rendre cohérente, il faudrait exporter les champs **LibelléDescrip** et **ValDescrip** dans une autre entité et les lier aux meubles pour éviter de répéter le nom du meuble à chaque fois qu’on lui ajoute une description.*

Ce même principe est repris dans la base de données de Koha pour gérer le format MARC (Voir paragraphe E.1.6.3 à la page 106).

E.1.6.3 Les méta données dans Koha

Étant donné qu'il y a des différences entre la structure des champs de notices aux formats : MARC, UNIMARC, etc... Il est difficile de créer des tables avec un champ pour chaque tag de la structure MARC. Les développeurs de Koha ont appliqué le principe des métadonnées en encodant les TAG et sous champs du MARC comme des données dans une table.

Schématiquement, la table principale de la base de données MARC ressemble à ceci :

MARC_SUBFIELD_TABLE				
subfieldid	bibid	tag	subfieldcode	subfieldvalue
160	10	100	a	Erich Gamma
132	10	245	a	DESING PATTERNS
184	10	700	a	John Vlissides
183	10	700	a	Ralph Johnson
208	10	020	a	2711786447

Les champs :

subfieldid C'est une séquence numérique qui sert d'identifiant unique.

bibid L'identifiant unique de la table MARC_BIBLIO chargée de faire le lien avec la table BIBLIO

tag La liste des Tag du MARC21 utilisé dans cette notice.

subfieldcode Le sous champ correspondant au TAG.

100 Entrée principale des noms.

a Noms de l'auteur.

245 Titre

a Titre principal

700 Entrées additionnelles coauteurs

a Coauteurs

020 ISBN

a ISBN

subfieldvalue Valeur du sous champs MARC21

Il y a une occurrence par paire libellé(ex : 100 a) valeur (ex : Erich Gamma) dans la table MARC_SUBFIELD_TABLE. Pour une seule notice on va avoir autant d'occurrences dans la table qu'il y a de sous champs MARC utilisés.

C'est le champ "bibid" qui distingue les notices entre elle. Pour sélectionner cette notice, il suffit de demander au SGBD toutes les occurrences de la table MARC_SUBFIELD_TABLE dont le champ "bibid" vaut 10.

E.1.7 Clé primaire

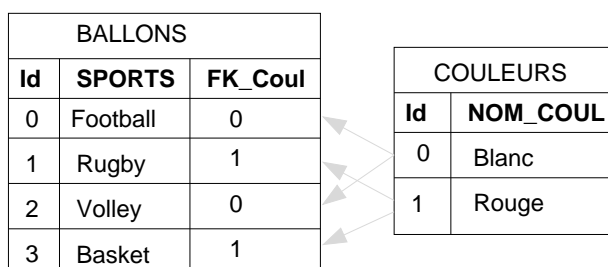
Une clé primaire et un champ ou un ensemble de champs servant à identifier de manière unique un enregistrement(tuple) dans une table.

La clé primaire sert aussi à lier les éléments de deux ensembles.

Prenons par exemple l'ensemble des ballons et celui des couleurs de ceux-ci.

BALLONS		
Id	SPORTS	COULEURS
0	Football	Blanc
1	Rugby	Rouge
2	Volley	Blanc
3	Basket	Rouge

La colonne Id est la **clé primaire** de la table BALLONS, sa séquence numérique identifie de manière unique chaque tuple(ligne) de la table. Mais une règle des bases de données relationnelles dit que les données doivent être uniques. Ici on répète les couleurs... La couleur ne fait pas partie de l'entité BALLONS. On va donc l'exporter vers une entité COULEURS.



La colonne Id de la table COULEURS est sa **Clé primaire**, elle permet en même temps de **lier** les couleurs aux ballons, en reprenant l'identifiant unique des couleurs dans la table BALLONS.

La colonne FK_Coul reprenant la clé primaire des couleurs ajoutées dans la table BALLONS est appelée **clé étrangère** (FK pour Foreign Key).

Ainsi on sait de quelle couleur sont les ballons sans la répéter en la réinscrivant en toute lettre. Cela ne rend pas la lecture plus difficile. Car par la suite, lorsqu'on veut afficher les ballons et leurs couleurs, on demande au système de gestion de base de données :

Sélectionne les colonnes SPORTS et NOM_COUL dans les tables BALLONS et COULEURS lorsque les champs FK_Coul (DE BALLONS) et id (de COULEURS) sont égaux.

Le SGBD² affichera ceci³ :

SPORTS	NOM_COUL
Football	Blanc
Rugby	Rouge
Volley	Blanc
Basket	Rouge

²Système de Gestion de Base de Données

³Il suit les flèches en gris

E.1.8 Chemin UNIX

E.2 Trucs et astuces

E.2.1 Monter une image ISO en loopback

Il n'est pas obligatoire de graver l'ISO du cdrom. Vous pouvez en faire l'économie en copiant l'image ISO sur le disque dur et en montant l'image ISO comme un périphérique via l'interface loopback.

Pour ce faire vous devez être connecté en tant qu'administrateur du système :

```
sh# mkdir -p /mnt/iso
sh# mount -o loop CDinstallKoha-07-09-2005-01.11.12.iso /mnt/iso/
sh# cd /mnt/iso/
```

La première commande sert à créer un répertoire qui servira de point de montage de l'image ISO ;

la deuxième monte l'image ISO sur le répertoire comme si il s'agissait d'un périphérique amovible ;

la troisième se déplace dans le répertoire qui se comporte exactement comme le cdrom si il avait été gravé et placé dans le lecteur.

Après cette étape, tous se passe comme si vous étiez sur le cdrom.

Un fois que l'installation de Koha sera terminée, il ne faudra pas oublier de démonter le faux périphérique.

```
sh# umount /mnt/iso/
```

E.2.2 Choisir un bon mot de passe

La qualité des mots de passe faciles à retenir mais difficiles à deviner est aussi un point très important. Une excellente façon de créer un mot de passe de qualité est de mémoriser un paragraphe de quelques lignes de son livre de chevet du moment, la citation d'un poème ou quoi que ce soit que l'on puisse retenir aisément. Ensuite on prend la première lettre de chaque mot en respectant majuscule et minuscule. Au moment de taper son mot de passe, il suffit de réciter mentalement ces quelques lignes, en tapant au clavier la première lettre que chaque mot.

Exemple :

Puisque ta voix étrange
Vision qui dérange
Et trouble l'horizon
De ma raison

Donne comme mot de passe : PtvéVqdEt1'hDmr

E.3 Notes destinées aux informaticiens

E.3.1 La table MARC_BIBLIO

Si les concepteurs de la base de données n'avaient pas eu besoin d'ajouter des champs de données dans la table MARC_BIBLIO tels que les dates, celle-ci n'aurait été qu'une CIF (Contrainte d'intégrité fonctionnelle) et cette table n'aurait pas existé. Mais ceci aurait eu pour inconvénient de placer une clé étrangère de BIBLIO dans la table MARC_SUBFIELD_TABLE alors qu'elle n'en dépend pas vraiment puisque MARC_SUBFIELD_TABLE reprend l'ensemble des TAG du MARC. Il y a une redondance des données mais pas de violation des règles de normalisation.

E.3.2 L'usage et la localisation des modules dans Koha

§KOHA est le répertoire d'installation de Koha

Les modules de Koha sont placés dans §KOHAintranetmodules qui est la racine⁴ des paquets.

On y trouve le répertoire C4 où sont rangés les modules appartenant à Koha.

Le chemin de la racine des modules est indiqué dans le fichier de configuration :

```
/etc/koha-httpd.conf
```

Par cette syntaxe :

```
SetEnv PERL5LIB "/usr/local/koha/intranet/modules"
```

Dans le code Perl ils sont appelés par la syntaxe :

```
use C4::Database;
```

Le chemin qui précède le répertoire C4 étant assigné à la variable "PERL5LIB".

E.3.3 Programme d'importation de données provenant de CDS/ISIS.

1. Sous DOS/Windows : migrer vers une version récente de CDS/ISIS permettant une exportation au format XML. Noter soigneusement le nom des champs correspondant aux éléments.
Par exemple : savoir que le champs "bidule" contient le nom de l'auteur
2. Récupérer ce fichier XML sous GNU/Linux et l'exploiter à l'aide d'un parseur XML. Un autre sous programme ira lire les tables "marc_tag_structure" et "marc_subfield_structure" de la base de données de Koha. Ces tables contiennent les TAG et sous champs du format MARC choisis lors de l'installation de Koha.

Deux méthodes sont envisageables pour le traitement du fichier XML :

SAX traite un fichier XML à la volée élément par élément (Attention l'entrée dans le contenu d'une balise peut être un élément).

Les avantages, inconvénients et spécificités de l'API SAX sont :

⁴Chemin relatif à partir duquel les paquets sont appelés

- Il permet de traiter des fichiers XML en pipeline venant par exemple d'un réseau. Le traitement s'effectue sans attendre la fin du transfert du fichier.
- Il peut traiter des fichiers très volumineux car il travaille élément par élément en déclenchant un évènement spécifique à la rencontre de chacun d'entre d'eux. Les éléments précédents sont déchargés de la mémoire.
- De part son principe de déclenchement actif d'évènements, il est particulièrement adapté à l'importation/exportation de données.

Par exemple, après avoir paramétré la table de correspondance entre les champs de CDS/ISIS et les TAG et sous champs du MARC, la rencontre de l'élément "bidule" dans le fichier XML provenant de CDS/ISIS déclenche un évènement, que l'on peut exploiter pour accomplir une action. Cette action peut être une exploitation directe de la donnée où son écriture dans un tube (cet aspect est décrit au paragraphe E.3.3.1)

DOM charge l'ensemble du fichier XML en mémoire et reconstitue l'arborescence de celui-ci de manière passive. Il incombe donc au programmeur d'écrire le code qui va parcourir le document XML et les actions correspondantes.

L'occupation en mémoire du fichier XML chargé par l'API DOM est environ de 3 fois le volume de celui-ci. Un fichier d'une taille de 1 Mo occupera 3 Mo de mémoire.

3. Insérer dans la base Koha les données extraites du fichier XML.

E.3.3.1 Importation des données provenant d'un fichier XML avec l'API SAX.

Prenons par exemple un document XML simplifié :

```
<CDS>
<livre>
<auteur>David Brownell</auteur>
<editeur>O'Reilly</editeur>
<titre>SAX2</titre>
</livre>
</CDS>
```

L'API SAX se base, entre autres, sur le principe du producteur et du consommateur.

Le producteur est le "parseur" du fichier XML, il est chargé d'alimenter le consommateur en évènement.

Le consommateur est chargé de gérer ces évènements et, par exemple, d'exploiter les données qu'il reçoit.

Le consommateur peut déléguer l'exploitation des données reçues en les écrivant dans un tube qui sera lu par un autre producteur.

```
Producteur -> Consommateur -> Tube -> Producteur -> Consommateur
```

Notre classe principale aura besoin de deux objets, un producteur et un consommateur :

```
import org.xml.sax;
import java. //Faire les importations nécessaires

XMLReader producteur;
//On crée une var de type CDSImport cette classe
//redéfinit des méthodes de DefaultHandler
CDSImport consommateur;
```

La classe `DefaultHandler` comporte des méthodes ayant une Implémentation par défaut. Il faudra en redéfinir trois :

```
public class CDSImport extends DefaultHandler{

    ....
    /* Déclaration des variables d'objet tel que :
    * Un booléen pour indiquer à characters()
    * ce qu'elle doit faire
    * Une table de hachage pour la correspondance
    * éléments/MARC
    * un tableau pour enregistrer les éléments
    * à ignorer.
    */

    startElement(String, String, String, Attributes){
        ...
    }
    characters(char[], int, int){
        ...
    }
    endElement(String, String, String){
        ...
    }
    ....
}
```

StartElement ()

Cette méthode lit les balises XML ouvrantes.

Exemple :

```
public void startElement(String uri,
String localName,
String qName,
Attributes attributes)
throws SAXException
```

Au moment où SAX lit la balise ouvrante `<titre>` le paramètre `qName` recevra la chaîne "titre".

endElement ()

Cette méthode lit les balises XML fermantes.

Exemple :

```
public void endElement(String uri,
String localName,
String qName)
throws SAXException
```

characters ()

Cette méthode lit le contenu textuel en omettant les balises et les attributs. Attention ! les retours à la ligne et les espaces d'indentation⁵ du fichier XML seront lus par cette méthode. "Receive notification of character data inside an element." Nous sommes bien à l'intérieur des éléments <titre>...</titre>

Exemple :

```
public void characters(char[] ch,
int start,
int length)
throws SAXException
```

La chaîne "ch" contiendra par exemple "SAX2" si la méthode characters() est appelée juste après que startElement() ai lu la balise <titre>.

E.3.3.1.1 Principe de fonctionnement de la classe CDSImport : SAX lit le fichier XML provenant de l'exportation des données gérées par CDS/ISIS.

Il y a trois possibilités pour établir une table de correspondance des champs CDS/ISIS avec les sous champs du MARC, par ordre décroissant d'interactivité :

1. lorsque la méthode startElement() rencontre un élément, le programme demande à l'utilisateur de lui faire correspondre un sous champ du MARC, ou de ne rien indiquer pour l'ignorer.
2. dresser dans un fichier texte, la liste des champs de CDS/ISIS auxquels la méthode startElement() doit réagir en demandant à l'utilisateur le sous champ MARC qui lui correspond ;
3. rédiger dans un fichier de configuration la correspondance des champs de CDS/ISIS et les sous champs du MARC, sous la forme :
auteur=100#a

Dans les deux derniers cas tous les champs qui n'auront pas été cités seront ignorés.

Prenons l'exemple du premiers cas :

La méthode startElement() rencontre un élément :

Le programme fait une recherche dans un tableau des éléments ignorés, si celui-ci est trouvé, il sera ignoré.

⁵ Indenter un code source c'est insérer des espaces ou des tabulations devant un bloc de code pour améliorer la lisibilité.

Si non, les clés de la table de hachage des éléments à traiter sont parcourues, si l'élément y est trouvé c'est qu'une correspondance a déjà été assignée à cette table de hachage. La clé étant l'élément du fichier XML sa valeur la clé primaire de l'enregistrement de la table `marc_subfield_structure`.

Cette table de la base de données de Koha est la référence des sous champs du format MARC utilisé.

Dans ce cas, la donnée encadrée par l'élément du fichier XML doit être utilisée par la méthode `characters()`.

Cette méthode est toujours appelée mais un booléen est affecté dans `startElement()` pour indiquer à `characters()` si il doit ignorer ou conserver la donnée qu'il reçoit.

Par exemple si l'élément `<titre>` est rencontré par la méthode `startElement()`, elle doit être lue par la méthode `characters()` qui recevra dans ce cas "SAX2".

```
<titre>SAX2</titre>
```

Si l'élément du fichier XML n'est trouvé ni dans la table de hachage des éléments à gérer ni dans le tableau des éléments à ignorer, le programme demande à l'utilisateur de lui faire correspondre un tag et un sous champ du MARC où de ne rien indiquer pour l'ignorer.

Chaque fois que la méthode `characters()` doit utiliser la donnée qu'elle reçoit, des enregistrements sont ajoutés dans la base de données de Koha avec les champs et sous champs du MARC décrivant cette donnée.

Ces ajouts sont regroupés dans une transaction SQL, se terminant par `COMMIT` ou `ROLLBACK`, selon que l'ensemble du fichier XML ait été lu avec succès ou non.

Bibliographie

- [1] Flanagan David. *JavaScript la référence*. Number ISBN 2-84177-212-8. O'Reilly, 2002.
- [2] Alligator Descartes and Tim Bunce. *Perl DBI le guide du développeur*. Number ISBN 2-84177-131-8. O'Reilly, 2000.
- [3] Eliotte Rusty Harold and W. Scott Means. *XML in a nutshell Manuel de référence*. Number ISBN 2-84177-223-3. O'Reilly, 2002.
- [4] Richard Helm, Erich Gamma, Ralph Johnson, and John Vlissides. *DESIGN PATTERNS Catalogue de modèles de conception réutilisables*. Number ISBN : 2-7117-8644-7. Addison Wesley, juillet 1999.
- [5] Matheron Jean-Patrick. *Comprendre Merise outils conceptuels et organisationnels*. Number ISBN 2-212-07502-2. Eyrolles, 1994.
- [6] Jose Morejon. *Merise par l'exemple*. Number ISBN 2-7081-1290-2 in Ingénierie des systèmes d'information. d'Organisation, 1991.

Index

- adresse IP, [8](#), [9](#)
- Apache, [17](#)
- architecture, [1](#)
- CDS/ISIS, [49](#)
- CGI, [17](#)
- Chemin UNIX, [108](#)
- chemin UNIX, [8](#)
- clé étrangère, [107](#), [109](#)
- Clé primaire, [107](#), [109](#)
- clé primaire, [29](#)
- Client, [16](#)
- crontab, [40](#), [99](#)
- distributions
 - Linux, [1](#)
- DocumentRoot, [103](#)
- GNU, [1](#)
- GNU/Linux, [1](#)
- Hôte, [102](#)
- Image ISO, [102](#)
- image iso, [6](#)
- Internaute, [16](#)
- internet, [102](#)
- ISO, [108](#)
- javascript, [18](#)
- Koha, [16](#)
- loopback (interface), [108](#)
- Méta données, [104](#)
- Méta données dans Koha, [106](#)
- métadonnées, [29](#)
- MARC
 - MARC21
 - TAG, [106](#)
 - UNIMARC, [106](#)
- Miroir, [102](#)
- modèle relationnel, [103](#)
- module, [17](#)
 - mod_perl, [17](#)
 - mod_php, [17](#)
 - module d'Apache, [17](#)
- Modules, [109](#)
- Mot de passe, [8](#)
- mysql, [8](#)
- Nom d'hôte, [8](#)
 - FQDN, [102](#)
 - Nom d'hôte complet, [8](#)
- OPAC, [102](#)
- PATH, [8](#), [108](#)
- Perl, [17](#)
- PHP, [17](#)
- PMB, [16](#)
- Port, [102](#)
- Protocole, [102](#)
- Règles de normalisation, [30](#), [103](#)
- Ressource, [102](#)
- Sauvegardes des données
 - backup, [34](#)
- Serveur, [16](#)
- serveur secondaire, [102](#)
- service, [102](#)
- système d'exploitation
 - GNU/Linux, [1](#)
- système d'exploitation., [1](#)
- Système de gestion bibliographique, [1](#)
 - SIGB, [1](#)
- URL, [102](#)
 - Uniform Resource Locator, [102](#)
- Web, [16](#)