A R Ã
Un système de gestion des miroirs sur Internet

RAPPORT D'ÉTAPE

Neilze Dorta
INRIA Rocquencourt, projet SOR
neilze.dorta@inria.fr

March 18, 1999

Table de matières

Table de figures

1 Introduction

Les systèmes à large échelle souffrent de problèmes de performance et de disponibilité dus aux grandes distances géographiques entre les sites, au grand nombre des sites et donc leur plus grande probabilité des fautes de communication, des pannes et de déconnexion. La réplication des données [?] dans des miroirs proches, peut permettre d'améliorer le temps d'accès et la disponibilité des données dans des systèmes de grande échelle. Les miroirs peuvent aussi permettre de réduire la charge des serveurs ainsi que le nombre des messages sur le réseau.

Pour que la réplication de données soit efficace, chaque utilisateur doit être capable de localiser et accéder aux copies proches de sa propre localisation. Les miroirs ont des problèmes dans le domaine de la localisation. La localisation des miroirs se fait manuellement par l'utilisateur. Cela entraîne à une perte de temps au moment d'accéder la donnée, paradoxalement à la raison d'être d'un miroir. On peut aussi citer le manque de motivation à accéder un miroir de la part de l'utilisateur. Si localiser un miroir peut être difficile, l'utilisateur préfererait, dans ce cas, accéder les données directement à la source, exigeant plus de bande passante et sacrifiant une meilleur performance.

Mon travail consiste en fournir un accès transparent aux miroirs. Ce travail se base sur la rédirection des requêtes http [?]. Ce service utilise une base de données des miroirs (BDM) et un proxy qu'interroge la BDM et redirige la requête si le miroir est connue de la dernière. La constitution de la BDM représente une grande partie du travail pour pouvoir fournir ce service. Un mécanisme de publication des miroirs est mis en oeuvre, pour assurer l'alimentation de la BDM.

2 Un problème intéressant et difficile

La localisation des réplicas dans un environnement à large échelle est difficile. Actuellement, les systèmes de nommage ne supportent pas la réplication des noms d'une façon dynamique et cohérente [?]. La liaison dynamique d'un seul nom avec plusieurs adresses ou l'identification des divers noms, liés à differentes adresses, comme étant le même document, sont des choses que l'on ne sait pas faire dans un environnement à large échelle. En plus, localiser la copie la plus proche (ou la meilleur copie) d'un document, exige des informations additionelles et des mise-à-jour fréquentes. Quelques solutions statiques permettent la localisation du 'meilleur' réplica [?, ?] mais ne supportent pas la découverte des ressources. Enfin, le système Globe [?] propose une solution dynamique et globale mais se heurte à des problèmes de stockage, de recherche et des mises à jour efficaces.

Dans le cadre de ce rapport, on s'intéresse particulièrement à la localisation des différentes copies, les difficultés pour découvrir l'existence d'un nouveau réplica ou simplement localiser le réplica le plus approprié. Ce rapport ne traite pas de la cohérence des objets mais juste de leur localisation. En plus, les solutions proposées focalisent sur les problèmes de localisation des miroirs, ne tennant pas compte des problèmes des caches.

Le miroir est la réplication d'une sous-arborescence entière des documents. Quand on a un miroir, les documents ont des noms différents par rapport à ceux de la source. Quand on a un ensemble des miroirs d'une même source, les documents ont des noms différents selon le miroir. Cela est dû au fait que la localisation des données fait partie des noms des documents sur Internet.

Localiser un miroir est difficile parce qu'il n'existe pas un service capable d'identifier de façon unique les miroirs correspondants à une même source. Un autre problème non résolu est de découvrir les nouveaux miroirs et de détecter la disparition de ceux déjà existants. Notre système propose la localisation des miroirs par coopération entre groupes d'organismes (voir figure 1). Nous entendons par organisme (voir figure 2) un groupe des sites coopérants, liés par un intérêt commun et géographiquement proches. Le groupe d'organismes coopérants sont eux aussi liés par des intérêts communs. La durée de vie d'un miroir étant assez longue, nous pouvons faire des choix de communication paresseuse entre les organismes, éliminant les contraintes du synchronisme dans un environnement large échelle. Les miroirs ont aussi la caractéristique de pouvoir être identifiés par un préfixe d'URL représentant toute leur arborescence. Cette proprieté diminue le volume des données sur la localisation à échanger entre les groupes d'organismes.

Dans un premier temps, notre solution se divise en deux parties : le support de localisation transparente des miroirs et le protocole de coopération inter-groupe pour la découverte des miroirs. La première partie sert à rendre transparent, vu de l'utilisateur, la localisation des miroirs; la deuxième partie permet de nourrir les bases de données de localisation de miroirs appartenant à chaque organisme. Ce document présente donc la spécification du support de localisation transparente des miroirs et ensuite celle du protocole de coopération inter-groupe pour la découverte des miroirs.

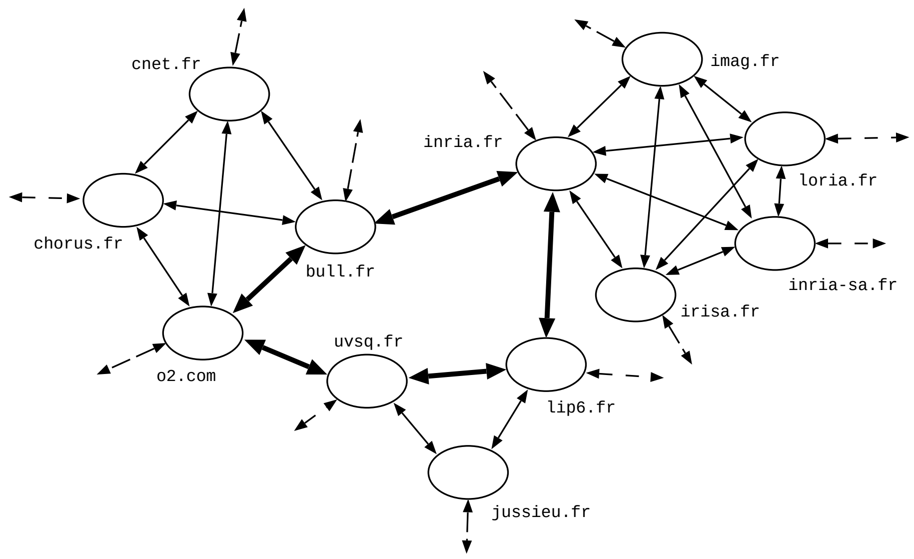

Figure 1: Groupes d'organismes

3 Support de localisation transparente des miroirs

Localiser un miroir peut être difficile et l'utilisateur préfererait, dans ce cas, accéder les données directement à la source, exigeant plus de bande passante et sacrifiant une meilleur performance. Le support de localisation transparente des miroirs vise à éliminer ce problème. Un usager pourra accéder au miroir le plus 'approprié' en ayant connaissance de l'URL source ou d'un de ses réplicas. Cet accès est totalement transparent et garantit un non retour sur le passé sur les documents déjà consultés.

L'accès à différentes copies peut genérer des incohérences dans la vue de l'utilisateur. Le service de localisation transparente des miroirs assure le non retour sur le passé au client. Le client pourra accéder diverses copies mais ne verra jamais une version de la donnée plus ancienne que celle déjà vue.

La localisation transparente des miroirs est un service qui permet à l'utilisateur d'accéder à une copie de la donnée sans connaître sa localisation. Dans la figure 3 on peut comprendre comment la localisation transparente des miroirs fonctionne:

- L'utilisateur fait une requête normale via un proxy;

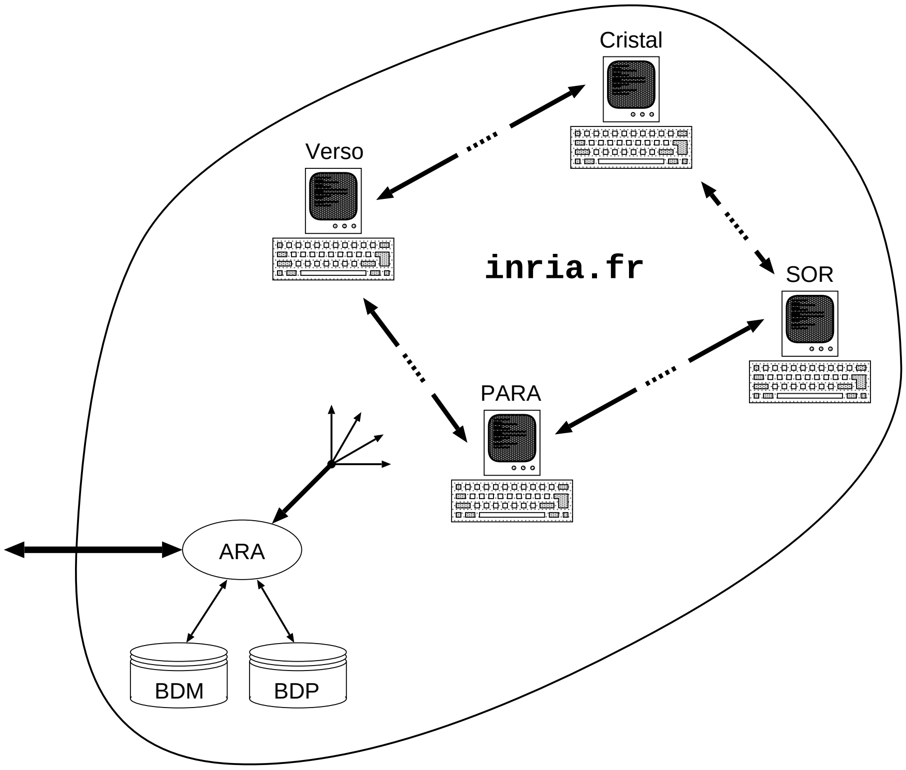

Figure 2: Un organisme

- Le proxy interroge le service de localisation : ARA;
- ARA redirige vers la meilleure localisation.

L'URL fourni par l'utilisateur peut être l'URL du document source ou l'URL d'une copie de ce document. Dans les deux cas, le support est capable d'établir la correspondance et associer l'URL donnée à l'URL copie la plus appropriée.

3.1 La base de données de miroirs (BDM)

L'accès transparent aux miroirs passe par la création d'une base de données des divers miroirs liés à la même source. Le préfixe de la sous-arborescence de l'URL source est donc l'identificateur unique des tous les miroirs répliqués.

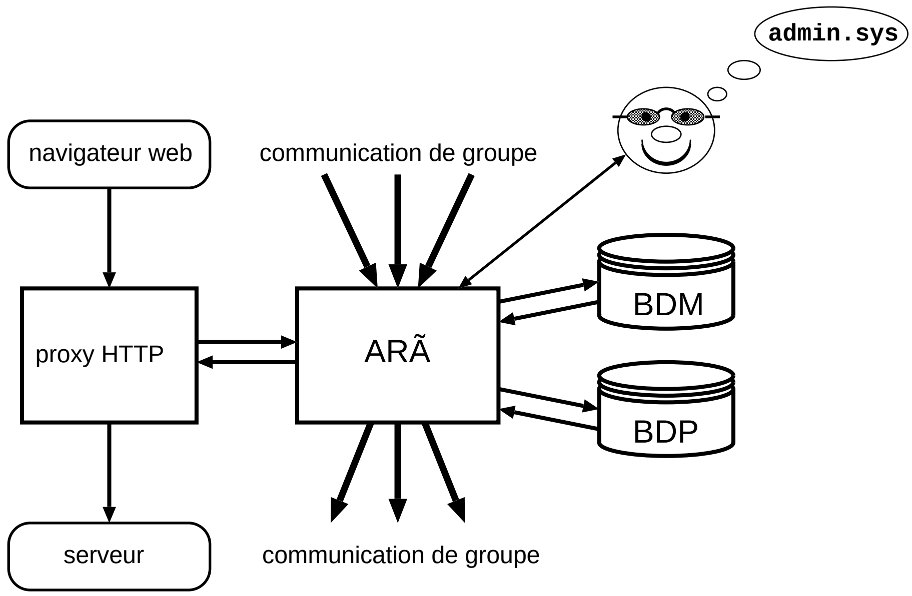

Figure 3: L'architecture ARA

Pour chaque serveur 'miroré' on établit une liste de préférence des miroirs ainsi que la date de sa dernière mise-à-jour.

C'est cette base de données qui permet la recherche des miroirs par rapport à une URL donnée. Il y a une base de données de miroirs (BDM) par organisme. Nous entendons par organisme un groupe des sites coopérants, liés par un intérêt commun et géographiquement proches. Le groupe d'organismes coopérants sont eux aussi liés par des intérêts communs. Chaque organisme échange des informations de sa base de données avec les autres de façon asynchrone et paresseuse. L'échange d'information se fait donc de façon épidemique entre organismes et entre groupes d'organismes.

3.2 L'interface d'accès

L'interface d'accès de la base de données de miroirs (BDM) permet la consultation de cette base et sa modification. Consulter la base pour identifier l'existance d'un miroir, pouvoir ajouter un miroir afin de nourrir la base de données, modifier le contenu d'un miroir afin d'avoir sa date de dernière mis-à-jour ou effacer un miroir à la suite de sa disparition ce sont les services offerts par cet objet.

identificateur unique	identificateur miroir	*best*	version
prefixe de l'URL source de la sous-arborescence	prefixe de l'URL miroir de la sous-arborescence	entier	date de la derniere mise-a-jour
source 1	miroir 1	n1	date 1
	miroir 2	n2	date 2
	miroir 3	n3	date 3
	⋮	⋮	⋮
source 2	miroir 1	n1	date 1
	miroir 2	n2	date 2
	miroir 3	n3	date 3
	⋮	⋮	⋮
⋮	⋮	⋮	⋮
source n	miroir 1	n1	date 1
	miroir 2	n2	date 2
	miroir 3	n3	date 3
	⋮	⋮	⋮

Figure 4: Base de données de miroirs

La BDM peut être accédée de trois façons différentes : par l'administrateur système, par un autre organisme coopérant et par le proxy quand le *browser* émêt une requête. L'administrateur système peut invoquer toutes les méthodes de l'interface. Les autres organismes coopérants accédent indirectement à l'interface par l'intermédiaire d'un server et l'entité ARA local. Ces dernières, ne peuvent pas faire de mise-à-jour sur la métrique (liste de préférences des miroirs) d'un miroir; ce concept étant propre à chaque organisme en particulier. Le *browser* a accès à l'interface BDM à travers d'un HttpProxy et d'une entité ARA local. Ces objets assurent à l'usager un accès transparent aux miroirs repértoriés dans la BDM local; les méthodes utilisés étant ceux qui consultent le contenu de la BDM seulement.

3.2.1 Comment nourrir la base de données

- Interactive

  La base de données peut être nourrie de façon interactive. L'administrateur système peut créer une entrée pour un nouveau miroir ainsi qu'utiliser toutes les fonctionalités offertes par l'interface.

- Coopération

  Une coopération inter-organismes (décrite dans le chapitre suivant) permet au divers BDM de s'échanger des informations sur les miroirs afin de découvrir de nouveaux miroirs et de mettre à jour les informations sur les miroirs déjà catalogués.

3.2.2 Les méthodes

Les méthodes d'accès à l'objet base de données sont les suivants:
int ***addMirror***(MirrorID mid)
int ***delMirror***(MirrorID mid)
int ***delSource***(String sup)
int ***updateMirrorVersion***(MirrorID mid, Date d)
int ***updateMirrorVersion***(String urlp, Date d)
int ***updateMirrorMetric***(String urlp, int m)
MirrorID ***findMirrorByPrefix***(String urlp)
MirrorID ***findBestMirrorByPrefix***(String urlp)
MirrorIDList ***findSource***(String urlp)
MirrorIDList ***lookup***(String urlp)

3.2.3 int *addMirror*(MirrorID mid)

La méthode *addMirror* rend possible l'ajout d'un objet miroir 'mid' à la base de données. Tout miroir doit être forcément lié à un préfixe d'url source, l'identificateur unique de l'ensemble des réplicas de cette sous-arborescence. Cette méthode retourne un entier : zero si tout s'est bien passé et -1 sinon.

3.2.4 int *delMirror*(MirrorID mid)

La méthode *delMirror* permet d'effacer un miroir après sa disparition. L'objet miroir 'mid' est identifié par son préfixe d'url. Cette méthode retourne un entier : zero si tout s'est bien passé et -1 sinon.

3.2.5 int *delSource*(String urlp)

La méthode *delSource* permet d'effacer une liste de miroir correspondant à la même entrée dans la BDM. Cette méthode sert à éliminer une source après sa disparition. L'entrée de la BDM est identifiée par un préfixe d'url, soit d'un de ses miroirs soit de la source elle même. Cette méthode retourne un entier : zero si tout s'est bien passé et -1 sinon.

3.2.6 int *updateMirrorVersion*(MirrorID mid, Date d)

L'appel *updateMirrorVersion* permet le changement de l'information liée à la version du miroir. L'objet miroir 'mid' identifie le miroir à modifier. Si la date 'd' passée en paramètre est plus récente que la date dans la base de données, alors la date est modifiée. Cette méthode retourne un entier : zero si tout s'est bien passé et -1 sinon.

3.2.7 int *updateMirrorVersion*(String urlp, Date d)

L'appel *updateMirrorVersion* permet le changement de l'information liée à la version du miroir. Le préfixe url du miroir 'urlp' identifie le miroir à modifier. Si la date 'd' passée en paramètre est plus récente que la date dans la base de données, alors la date est modifiée. Cette méthode retourne un entier : zero si tout s'est bien passé et -1 sinon.

3.2.8 int *updateMirrorMetric*(String urlp, int m)

L'appel *updateMirrorVersion* permet le changement de l'information liée à la métrique du miroir. Le préfixe url du miroir 'urlp' identifie le miroir à modifier. La métrique, qui représente le meilleur miroir de la même source, entraîne la redéfinition de toutes les autres métriques de la même entrée. Cette méthode retourne un entier : zero si tout s'est bien passé et -1 sinon.

3.2.9 MirrorID *findMirrorByPrefix*(String urlp)

La méthode *findMirrorByPrefix* permet de trouver un miroir par une url passée en paramêtre. Cette url peut correspondre au préfixe url de la source ou d'un quelconque miroir. Le miroir retourné est le premier de la liste si le préfixe d'url passé correspond à celui de la source ou le miroir lui même sinon. Cette méthode retourne un MirrorID si le miroir est trouvé, sinon elle retourne 'null'.

3.2.10 MirrorID *findBestMirrorByPrefix*(String urlp)

La méthode *findBestMirrorByPrefix* permet de trouver le meilleur miroir pour une url passée en paramètre. Cette url peut correspondre au préfixe url de la source ou d'un quelconque miroir. Le miroir retourné est celui qui a la meilleur métrique selon la définition choisie. Cette méthode retourne un MirrorID si le miroir est trouvé, sinon elle retourne 'null'.

3.2.11 MirrorIDList *findSource*(String urlp)

L'appel *findSource* permet de trouver tous les miroirs correspondants à une source par un préfixe d'url passé en paramètre. Ce préfixe d'url peut correspondre à celui de la source ou d'un quelconque miroir. Cette méthode retourne une liste des miroirs correspondant à la même entrée de la base de données. Si l'url n'est pas connu de la BDM, un 'null' est retourné.

3.2.12 MirrorIDList *lookup*(String urlp)

La méthode *lookup* permet la localisation dans la base de données d'un miroir contenant l'url connu de l'utilisateur. Cette méthode retourne une liste des miroirs correspondant à la même entrée de la base de données. Si l'url n'est pas connu de la BDM, un 'null' est retourné.

4 Protocole de coopération inter-groupe pour la découverte et publication des miroirs

Le protocole de coopération inter-groupe,basé sur le protocole Relais [**?**], permet à chaque organisme du publier les miroirs dont il a connaissance, et en contrepartie de découvrir les miroirs publiés par les autres organismes partenaires. Pour cela, plusieurs fonctionnalités sont fournies à travers un ensemble de messages qui forment le protocole lui même. En plus des messages, une gestion de groupe est nécessaire pour rendre possible une coopération dynamique et efficace. Dans ce sens, une base de données contenant les organismes coopérants pour chaque organisme en particulier doit exister. Cette base de données doit être mise-à-jour de façon dynamique et permettre une indépendance pour chaque organisme par rapport à son groupe de coopération. La figure en dessous montre la coopération dans un groupe fermé et fixe. Par la suite, la question de la formation des groupes ainsi que la possibilité de participer à des groupes différents sera discutée et detaillée.

4.1 Les messages du protocole

Le protocole est formé par un ensemble de messages contenant des informations nécessaires à la découverte des miroirs, leur disparition et leur mise-

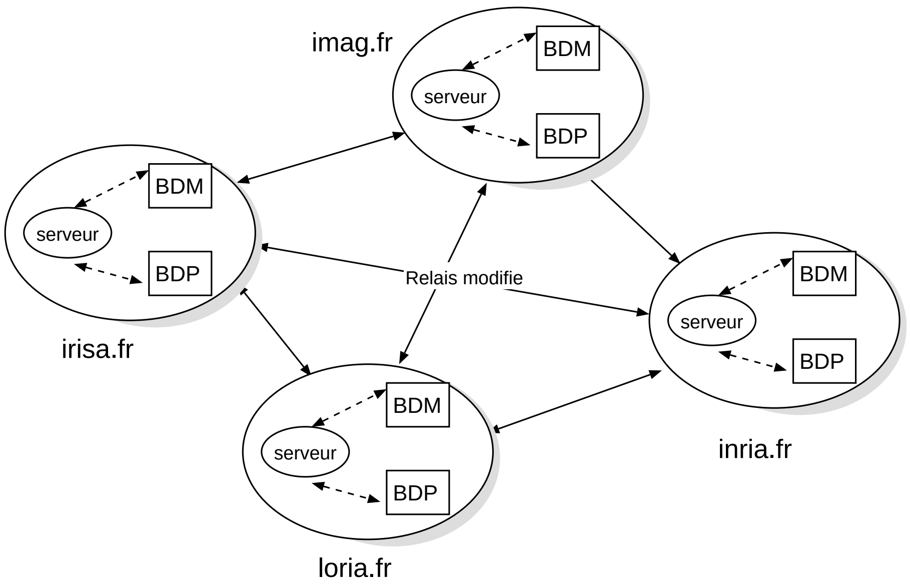

Figure 5: Coopération inter-organismes

à-jour. L'échange d'information peut se passer de façon passive ou active. On entend par échange passif celui où on ne cherche pas à être informé; l'information sera acquise si un des organismes avec qui on coopére l'envoie. Au contraire, l'échange actif exige une démarche en quête de l'information.

Les messages du protocole offrent une interface minimale pouvant informer de l'ajout d'un miroir (source), son retrait ainsi qu'une mise-à-jour d'un miroir. L'interface permet de demander aux organismes coopérants s'ils connaissent un miroir en particulier ou des miroirs d'une certaine source. Enfin, il est aussi possible de faire une requête sur l'existance des miroirs par rapport à une URL donnée.

4.1.1 Formats des messages

ADD_MIRROR
int *addMirror*(MirrorID mid)
DEL_MIRROR
int *delMirror*(MirrorID mid)
DEL_SOURCE
int *delSource*(String sup)
UPDATE_MIRROR_VERSION_BY_PREFIX
int *updateMirrorVersion*(String urlp, Date d)
UPDATE_MIRROR_VERSION_BY_MIRROR
int *updateMirrorVersion*(MirrorID mid, Date d)
FIND_MIRROR_BY_PREFIX
MirrorID *findMirrorByPrefix*(String urlp)
FIND_SOURCE
MirrorIDList *findSource*(String urlp)
LOOKUP
MirrorIDList *lookup*(String urlp)

4.1.2 ADD_MIRROR : int *addMirror*(MirrorID mid)

Le message ***ADD_ MIRROR*** notifie à tous les organismes qui coopérent avec l'organisme local la découverte d'un nouveau miroir 'mid'. Un organisme reçoit le message ***ADD_ MIRROR*** et l'insére dans sa propre BDM. Si le miroir en question est déjà connu de sa BDM, on vérifie la date de sa dernière mis à jour et la modifie si celle-ci est plus récente. Ensuite, chaque organisme répond avec un acquittement à l'organisme qui a envoyé la notification.

4.1.3 DEL_MIRROR : int *delMirror*(MirrorID mid)

Pour informer du retrait d'un miroir, l'organisme envoie le message ***DEL_ MIRROR*** à tous les organismes avec qui il coopére. L'identificateur du miroir à supprimmer est passé en paramètre : 'mid'. L'organisme qui reçoit le message vérifie si ce miroir fait partie de sa BDM. Si oui, on compare la date de la dernière mise-à-jour avec la date de la BDM, si celle-ci est plus récente que celle de la BDM le miroir est éffacé. Ensuite, chaque organisme répond avec un acquittement à l'organisme qui a envoyé la notification.

4.1.4 DEL_SOURCE : int *delSource*(String urlp)

besoins et les objectifs pour une telle interface

Le protocole permet le retrait de tout un ensemble des miroirs correspondant à la même source, soit le retrait de la source elle même par le message *DEL_ SOURCE*. Pour cela, le préfixe de la sous-arborescence de la source est envoyé 'urlp' (ou le préfixe d'un de ses miroirs) à l'intérieur du message à tous les organismes avec qui il coopére. Les organismes à leur tour effaceront cette entrée de la BDM après avoir comparé la date de la disparition de la source avec la date de la création de la source local. Ensuite, chaque organisme répond avec un acquittement à l'organisme qui a envoyé la notification.

4.1.5 UPDATE_MIRROR_VERSION_BY_MIRROR : int *updateMirrorVersion*(MirrorID mid, Date d)

Lors d'une mise-à-jour d'un miroir, le message *UPDATE_ MIRROR_ VERSION_ BY_ MIRROR* envoie le miroir 'mid' et la version 'd' à tous les organismes avec qui une coopération a été établie. Les organismes à leur tour mettent à jour la date de la dernière mise-à-jour si celle-ci est plus récente que celle connue de leur BDM. Ensuite, chaque organisme répond avec un acquittement à l'organisme qui a envoyé la notification.

4.1.6 UPDATE_MIRROR_VERSION_BY_PREFIX : int *updateMirrorVersion*(String urlp, Date d)

Lors d'une mise-à-jour d'un miroir, le message *UPDATE_ MIRROR_ VERSION_ BY_ PREFIX* envoie le préfixe d'une url (source ou miroir) et la version 'd' à tous les organismes avec qui une coopération a été établie. Les organismes à leur tour mettent à jour la date de la dernière mise-à-jour si celle-ci est plus récente que celle connue de sa BDM. Ensuite, chaque organisme répond avec un acquittement à l'organisme qui a envoyé la notification.

4.1.7 LOOKUP : MirrorIDList *lookup*(String urlp)

Le protocole permet à un organisme de demander aux organismes coopérants s'ils ont dans leur BDM une entrée correspondant à l'URL d'un document passé à l'intérieur du message *LOOKUP*. La réponse à cette requête sera l'envoie de tout le contenu de l'entrée de la BDM : une liste des miroirs correspondant à tous les miroirs connus de cet organisme.

4.2 Les participants du protocole

Les participants au protocole sont des organismes formés par plusieurs sites géographiquement proches et liés par des intérêts communs. Nous appellons cet ensemble de sites des organismes. Ces organismes sont formés de façon explicite et s'identifient comme un organisme coopérant.

Les organismes ainsi formés peuvent coopérer entre eux. Les organismes qui ont des intérêts en commun et qui appartienent à une même région géographique sont les bons candidats à une coopération fructueuse. Les organismes partageant des intérêts communs ont de forte chance d'avoir une localité significative dans la recherche des documents sur le web. De même, les organismes bénéficiant d'une bonne liaison réseau auront un coût d'échange d'informations plus bas.

4.2.1 La base de données des participants

Chaque organisme ayant envie de coopérer avec d'autres organismes, doit connaître les participants de son groupe de coopération. Pour cela, une base de données contenant tous les organismes avec lesquels la coopération a lieu doit exister et être mise-à-jour. Cette base de données est particulière à chaque organisme ce qui permet à chaque organisme d'avoir son propre groupe de coopération différent des autres.

Cette base de données de participants permet au protocole de coopération de savoir l'adresse IP de chaque organisme (serveur) avec qui une colaboration a été établie. Les messages sont adressés à tout le groupe selon l'algorithme de filtrage choisi par chacun des participants.

4.3 La coopération inter-organismes

- La découverte des miroirs

  Nous abordons ce problème à travers la coopération entre plusieurs organismes. Un miroir inclut une sous-arborescence d'adresses. ARA tire profit de cette propriété qui dimunue le montant de données à connaître sur la localisation. L'idée principale est d'avoir un protocole coopératif entre organismes qui permet l'échange épidémique de metadonnées (information sur la localisation).

  À chaque fois qu'un organisme :

  découvre et repértorie un miroir,
  crée et repértorie un miroir ou

identificateur du groupe (entier)	adresse IP de l'organisme	nom de l'organisme	port IP de l'organisme
groupID 1	addrIP 11	name 11	portIP 11
	addrIP 12	name 12	portIP 12
	addrIP 13	name 13	portIP 13
	⋮	⋮	⋮
groupID 2	addrIP 21	name 21	portIP 21
	addrIP 22	name 22	portIP 22
	addrIP 23	name 23	portIP 23
	⋮	⋮	⋮
⋮	⋮	⋮	⋮
groupID n	addrIP n1	name n1	portIP n1
	addrIP n2	name n2	portIP n2
	addrIP n3	name n3	portIP n3
	⋮	⋮	⋮

Figure 6: Base de données de participants

est informé par ARA de l'existence d'un miroir et le repértorie,

tous les organismes avec qui l'organisme local coopére sont avertis de l'existence de ce miroir; sauf ceux qui ne veulent pas (un algorithme de filtrage pourra être mise en place) et celui qui font partie du même groupe d'organismes coopérants que l'émetteur du message.

- L'aspect épidemique de l'échange de l'information

L'information sur la localisation des miroirs peut être échangée d'une façon paresseuse. Le temps de vie d'un miroir est assez long pour justifier une propagation paresseuse de sa localisation. L'avantage d'une propagation paresseuse est le gain en bande passante.

Le protocole ARA utilise la communication épidemique point à point [**?**]. Ce choix se justifie dans la mesure que ce moyen de communication est tolérant aux fautes, ne nécessite pas la connaissance de tous les participants et facilite le support de la déconnexion.

L'échange d'information par épidemie est un modèle qui passe à l'échelle et profite de l'anatomie actuelle de l'internet. Cependant, cette méthode peut créer de cycles exigeant un contrôle sur les messages déjà reçues et envoyées, pour ne pas faire transiter des messages inutiles.

- La connaissance des nouvelles versions de miroirs

  La BDM du système ARA connaît pour chaque miroir la date de sa dernière mise-à-jour (version). Cette donnée est échangé entre les organismes coopérants. Quand un organisme prend connaissance d'une version plus récente d'un miroir, il fait connaître à ses partenaires l'existence de cette nouvelle version en leur transmettant le miroir correspondant.

  Cette connaissance des versions sur les miroirs peut être utilisée pour une éventuelle mise-à-jour des miroirs. Ces miroirs se mettraient à jour non pas en cherchant les données à la source mais en les cherchant sur un autre miroir : moins coûteux en latence et/ou bande passante et partageant ainsi la charge qui aurait eu le serveur de la source.

5 La Gestion de Groupe

5.1 Comment coopérer avec un autre organisme

La coopération entre organismes ne peut avoir lieu que si les organismes se connaissent entre eux. Chaque organisme doit connaître tous les autres organismes avec qui il maintient une rélation de coopération. La base de données de participants (expliqué avant) peut contenir les informations nécessaires, mais il faut la nourrir et la mettre à jour régulièrement.

Une gestion de groupe est donc nécessaire et l'objet responsable de cette gestion doit fournir une interface simple et complète qui permet aux organismes d'adhérer à un groupe, quitter un groupe, etc.

ARA contient un gestionnaire de groupe qui fournit les services présentés par la suite permettant le dynamisme et la flexibilité dans la coopération inter-organismes.

- s'abonner à un groupe

  Tout organisme voulant adhérer à un groupe, doit connaître un de ses participants. Un message ABONNER est alors envoyé à cet organisme. L'organisme contacté donnera ou non l'autorisation pour que l'abonnement aboutisse. Dans ce cas, les informations à propos du nouveau participant sont diffusées parmi les membres du groupe.

Ensuite, une procédure de conciliation avec le nouveau membre est déclanchée dans le but d'échanger les informations sur les miroirs connus de chacun. Toute la BDM est alors envoyée au nouveau membre qui à son tour fera le tri dans sa propre BDM avant d'envoyer à chaque membre du groupe ses informations, d'une façon paresseuse et par une comunnication point à point.

- se désabonner d'un groupe

  La procédure de désabonnement consiste à envoyer le message DESABONNER à tous les membres du groupe. Cette procédure s'assure qu'aucun message est en transit. Ensuite chaque organisme supprimme le participant en question de leur base de données de participants (BDP).

- coopérer avec plusieurs groupes

  Un organisme peut coopérer avec plusieurs groupes. Pour cela, dans sa BDP il aura une entrée pour chaque groupe distinct. La coopération avec plusieurs groupes vient du fait qu'un organisme s'intéresse à des axes diverses et qu'un seul groupe ne peut couvrir tous les domaines de son intérêt.

  La coopération entre plusieurs groupes a l'avantage de passer à l'échelle. L'idée principale étant d'avoir un protocole de coopération entre groupes d'organismes qui permet l'échange épidemique de meta données. Ces données de localisation des miroirs seront propagées au fur et à mesure que l'épidemie croise les frontières des groupes.

  La coopération entre plusieurs groupes entraîne un système de filtrage d'information pour éviter l'innondation des messages par cycles ou par le biais des groupes qui ne s'intéressent pas à l'information. Un système de sécurité est aussi fondamental étant donné la circulation 'libre' des messages entre les groupes.

6 Travail de recherche future

6.1 La cohérence

6.1.1 La cohérence d'un miroir avec sa source

La cohérence d'un miroir avec sa source se fait de deux façons : soit le miroir demande à la source s'il y a une nouvelle version ou la source connaît ses miroirs et les avertit dès qu'une nouvelle version est disponible. La première solution peut générer des messages inutiles; la seconde suppose la connaissance des tous les miroirs.

La notion de version existe si la source est consciente d'être 'miroré'. Les versions sont crées après que des modifications significatives ont été éffectuées évitant des mise-à-jour sur des modifications minimes. Dans la réalité, n'importe quelle source peut être 'mirorée' et ne gérer aucun système de versions.

Une autre solution possible pour le maintien de la cohérence est de faire coopérer les miroirs d'une même source. Les miroirs se connaissent (en utilisant le système de coopération décrit avant) et s'échangent des messages entre eux notifiant l'existence d'une nouvelle version chez la source.

Une autre approche est celui de profiter de la coopération entre les organismes (dans le modèle décrit avant) pour notifier l'existence de nouvelles versions. Un miroir qui se met à jour peut informer les autres miroirs de la version actuelle en utilisant le protocole de coopération entre les organismes pour cela.

6.1.2 Le non retour sur le passé

Le problème du retour sur le passé se pose lorsqu'un miroir devient inaccéssible. L'utilisateur qui est en train d'accéder à ce miroir aura une erreur et cherchera à relancer la requête. Le problème surviendra si le miroir alors connecté possède une version plus ancienne que celle que l'utilisateur avait déjà consultée.

ARA assure à ses utilisateurs le 'non retour sur le passé' pour chaque document consulté. Cela veut dire que l'utilisateur peut se trouver avec une copie différente de celle qu'il a consulté auparavant, mais cette nouvelle copie est sûrement plus récente que la précedente.

Le 'non retour sur le passé' peut être assuré de deux façons :

- Chercher directement à la source

  Dès que le problème survient, l'utilisateur est aiguillé directement à la source. Cette dernière ayant surêment la version la plus récente.

- Connaître la dernière version de tous les miroirs accédés par chaque utilisateur

  La connaissance de toutes les versions accédées par chaque utilisateur, permet le choix d'un autre miroir qui a une version équivalente ou plus récente que celle accédée auparavant.

Cette solution requiert le stockage de la dernière version de tous les miroirs accédés par chaque utilisateur. Cela peut causer des problèmes de place en mémoire ou en disque. D'un autre côté, le *meilleur* miroir peut être choisi lors d'une déconnéxion imprévue. Cela implique un éventuel gain en latence et en bande passante.

6.1.3 La mise-à-jour des versions des miroirs

La mise-à-jour des versions des miroirs se fait par le miroir lui même ou par le miroir après que la source lui a signalé l'existance d'une nouvelle version (comme expliqué avant).

Dès qu'un miroir se met à jour et cette information est propagée sur le 'réseau ARA', la connaissance d'une nouvelle version peut être utilisée pour une éventuelle mise-à-jour des miroirs. Les miroirs se mettraient à jour non pas en cherchant les données à la source mais en les cherchant sur un autre miroir : moins coûteux en latence et/ou bande passante et partageant ainsi la charge qu'aurait eu le serveur de la source.

6.2 Le filtrage de l'information

Le partage de l'information par coopération a l'avantage de sélectionner naturellement les informations selon le groupe avec qui la coopération a lieu. Cependant, dans un modèle de coopération où il existe des intersséctions entre les groupes, certaines informations ne sont pas nécessairement intéressantes. Dans la section qui suit nous discutons les problèmes liés à l'intérsection entre les groupes et le filtrage nécessaire de l'information.

- Boucle de messages

  Supposons plusieurs groupes qui coopèrent entre eux. Un groupe coopère avec une partie des autres groupes et les autres groupes font de même. Les super-groupes de coopération ont des intérsections entre eux. La figure en dessous montre ce scénario. Les intersections peuvent finir par former des cycles dans le graphe. Les cycles peuvent générer des messages redondants et ainsi dégrader la performance du protocole et du réseau.

- Solution simple : Degré de Contamination

  Une façon de filtrer les informations à échanger avec les autres groupes est donc d'établir un *degré de contamination*. Nous entendons par *degré de contamination* une métrique pour savoir jusqu'à quel niveau d'indiréction sur la coopération, l'information est accéptable.

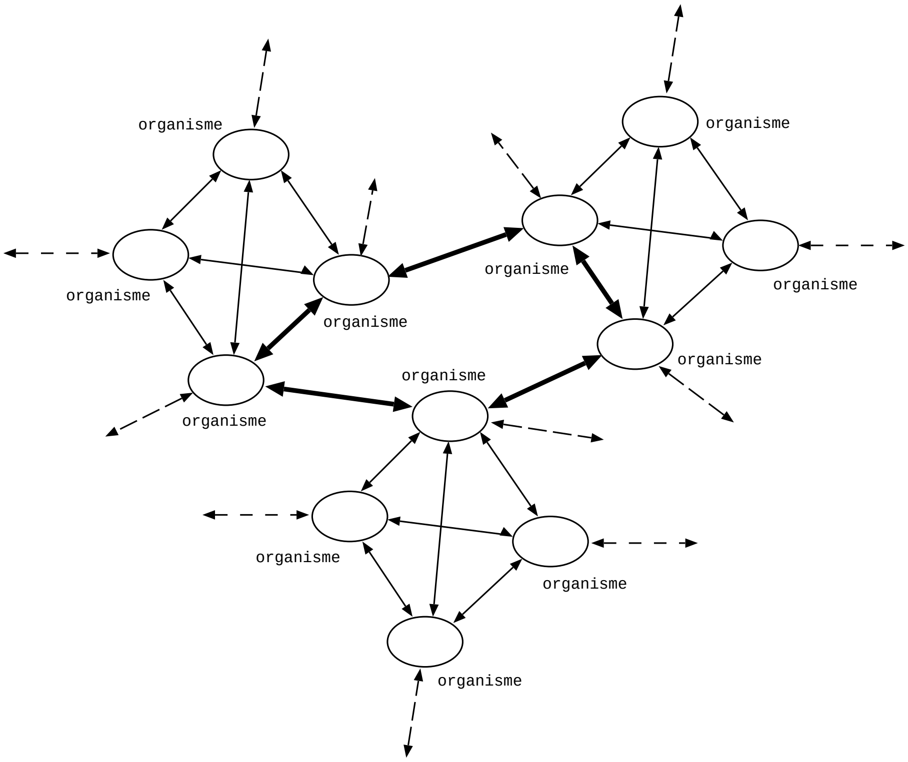

Figure 7: L'existence des cycles entre groupes d'organismes

Cette solution a l'avantage d'être simple. Si les groupes de coopération ont une tendance naturelle à chercher la localité, cette solution doit en plus fournir de bons résultats.

- Solution elaborée : Mots-clé, annotation...

  Une solution plus élaborée serait de prendre en considération les intérêts de chaque organisme [**?**]. Cela pourrait se faire en attribuant à chaque réplica (miroir) un ensemble des mots-clé. L'idée principal étant de prendre en compte la sémantique des intérêts de chacun, indépendament de la façon de le faire.

  Ce type de solution est difficille à mettre en oeuvre et ne garantit guère des meilleurs résultats, à l'exception de faire un travail exaustif sur le catalogage de mots-clé ou sur autres formes d'identification des intérêts.

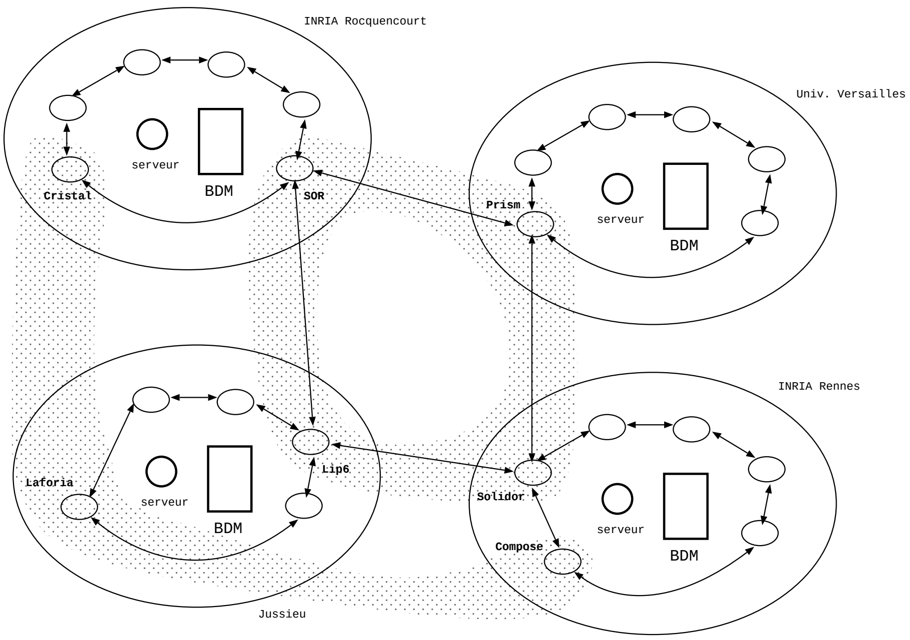

Figure 8: L'existence des cycles entre sites de groupes d'organismes

6.3 Désigner le *meilleur* miroir

Dans le modèle de découverte des miroirs, un problème de grand intérêt est celui de savoir choisir le *meilleur* miroir. Le *meilleur* miroir obéit à différents critères et, savoir toujours quel miroir est le *meilleur* requiert une mise-à-jour constante par rapport au critère choisi. Tout cela a un coût à comparer aux avantages de connaître le *meilleur* miroir au moment de la requête.

- Les différents critères

  Différents critères peuvent être retenus pour définir le *meilleur* miroir. Le choix va dépendre de la priorité des besoins de chaque organisme. Nous pouvons énumerer quelques critères qui nous semblent intuitivement pertinents dans le cadre du problème étudié :

  - le miroir qui offre l'accès le plus rapide;
  - le miroir qui a la version la plus récente;
  - le miroir qui est le moins demandé (accédé);

- ...

- La mise-à-jour constante de l'information

  Quelque soit le critère choisi, les paramètres qui définissent le critère sont variables dynamiquement. Cela implique que les modifications de ces paramètres doivent être pris en compte régulièrement de façon à ce que le système fournisse une information correcte. Le problème se pose donc sur le compromis à prendre entre le gain effectué en accédent le *meilleur* miroir et le coût à payer pour obtenir les informations nécessaires.

- Le coût à payer et les avantages

  Une étude précise est nécessaire pour comparer le gain d'un accès optimisé par rapport au coût pour obtenir les informations dont on a besoin. Le résultat devrait prendre en considération un grand nombre de paramètres. Par exemple, la localisation des miroirs, le réseau en question, la surcharge du système, etc.

6.4 Accès à l'information par un individu

Le modèle de coopération entre organismes ne couvre pas le besoin d'un individu à accéder les informations concernant la localisation des miroirs. Un individu ne faisant pas partie d'un organisme, les informations sur la localisation des miroirs ne l'arrivent pas. Il serait intéressant d'avoir une solution spécifique destinée aux individus non agréés à un organisme. Dans ce sens, deux solutions sont envisageables.

- Centraliser et publier l'information

  Accès à l'information par un organisme qui recueille et publie l'information. Un organisme coopére avec d'autres organismes, récolte des informations sur la localisation des miroirs et met ces informations à la disposition. Les informations sont ainsi publiées et accéssibles à tous. Cet organisme doit se faire connaître comme un centre d'information sur la localisation des miroirs.

- Accès descentralisé : coopération unilaterale

  Accès à l'information par coopération unilateral avec des organismes. Les organismes qui le souhaitent, donnent accès à leur BDM à tous ceux qui veulent la consulter. L'accès à l'information se fait par une coopération unilaterale : l'organisme fournit l'information à la demande et ne récupère aucune information en retour.

6.5 L'intéraction avec le cache

Afin de profiter de tous les réplicas présents dans notre système, il faudrait prendre en compte les réplicas qui se trouvent dans les caches aussi. Ainsi, faut-il voir que tout système futur de localaistion doit à tout prix inclure dans sa conception les caches (voir figure 9).

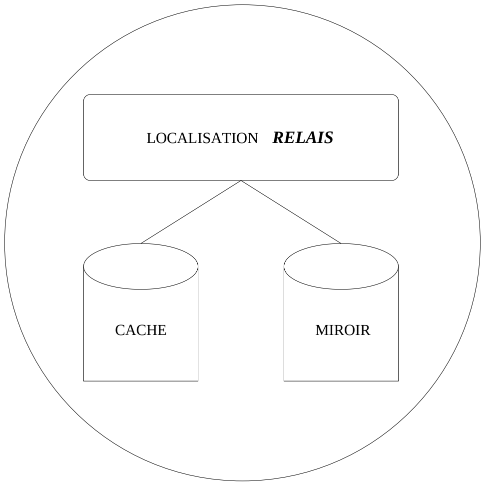

Figure 9: Système de localisation global

Les différences majeures entre caches et miroirs peuvent se résumer en deux points :

1. une durée de vie limitée, soit par besoin de cohérence ou par besoin d'aménagement d'espace.
2. les caches ont une politique de réplication directionnée par les requêtes de leurs clients, alors que les miroirs sont plutôt des entités de réplication guidées par les changements sur les serveurs eux mêmes.

De ce fait, il parait plus logique que les caches soient plus proches des clients que les miroirs (sauf pour les cas speciaux des miroirs montés dans l'organisme lui-même).

Ainsi, aboutissons-nous à une suggestion de système final de localisation qui englobe la coopération entre caches et miroirs (voir figure **??**).

La requête d'un client passe toujours par le cache de l'organisme ou se trouve ce dernier. De ce fait,si le cache est capable de satisfaire la requête

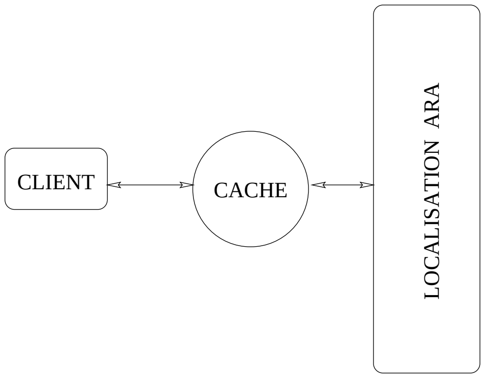

Figure 10: L'intéraction avec le cache

le client est servi directement. Sinon, le cache redirige la requête vers ARA qui retourne à son tour l'URL (modifiée ou pas) au cache. Ce dernier alors, procede par une inspection locale pour voir si cette nouvelle URL (dans le cas ou elle a été modifiée) est dans son cache ou pas. Dans le premier cas, le client est servi directement, sinon, la nouvelle URL est utilisée par le cache pour aller chercher le nouveau document.

Une amélioration de ce système serait le retour par le système ARA d'une liste d'URLs correspondant à l'URL demandée par le client. Ainsi, le cache pourra-t-il voir s'il a déjà une copie de l'une des URLs valide dans son cache et la servir directement au client.

Ces deux dernières solutions peuvent aussi être améliorées par un système orthogonal de QOS qui peut venir s'incruster au niveau de la localisation globale pour permettre une meilleure gestion de la localisation avec contraintes exigées par le client.

6.6 La recherche par préfixe

La recherche par préfixe est un point fondamentale pour le bon fonctionnement du système ARA. Pour une côté, le traitement par préfixe permet l'identification (nommage) de toute une sous-arborescence. Cela facilite le stockage des données de localisation et améliore le temps de recherche d'un miroir dans sa base de données. D'un autre côté, le système de préfixe permet de représenter les miroirs et les sous-miroirs qui peuvent exister éventuellement (par exemple, on a des miroirs de tout MIT et on a aussi des miroirs reférents à une partie des données du MIT). La recherche par

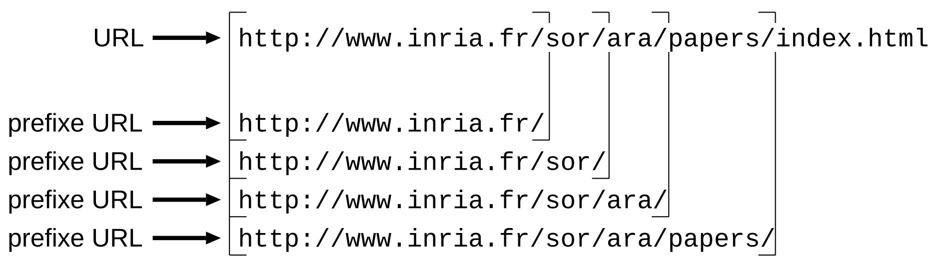

Figure 11: Un exemple de préfixe

préfixe peut se faire par comparaison séquentiel, par hachage des préfixes successifs ou par un arbre binaire lexicographique.

6.6.1 Recherche par arbre binaire lexicographique

Un arbre binaire lexicographique consiste en un arbre ordonné selon l'ordre alphabétique. Tous les noeuds qui se trouvent à droite sont lexicographiquement plus grand (soit alphabétiquement ils viennent après). Les noeuds à gauche représentent les données alphabetiquement placées avant. Le caractère binaire de l'arbre donne choix à que deux chemins à la fois. Dans notre système chaque noeud représent un préfixe et contient une lite des préfixes racrochée à lui. Cette liste peut être pointée par plus d'un noeud ce qui permet une recherche de la source vers le miroir et d'un miroir vers la source. En plus, les sous-miroirs peuvent être trouvés; jusqu'à celui qui a le plus grand préfixe satisfaisant la recherche.

L'arbre binaire lexicographique balancé permet une recherche precise et efficace sur les préfixes. Cette recherche permet la localisation de tous les préfixes et sous-préfixes liés entre eux. Cela permet au système de localisation transparente de localiser tous les miroirs pouvant répondre à la requête émise. Le coût de cette recherche est dans le pire de cas égal au coût du parcours de l'hauteur de l'arbre. Des algorithmes spécifiques aux arbres binaires permettent un équilibrage de l'arbre qui garanti une hauteur minimal de l'arbre en question.

7 Conclusion

Le système ARA fournit une gestion des aliases pour les miroirs ainsi qu'un mécanisme de publication et découverte des miroirs. Ces services sont disponibles à travers d'un support d'accès transparent aux miroirs et un protocole de coopération paresseuse entre groupes d'organismes que s'échangent des informations sur la localisation des miroirs. ARA ouvre des nouvelles possibilités pour améliorer le maintient de la cohérence entre les miroirs d'une même source. Toutes ces solutions peuvent présenter un gain en bande passante et en performance, ainsi que dans la répartition de la charge et une meilleure cohérence.

Le modèle de localisation des miroirs présenté est une solution non globale mais qui supporte l'environnement large échelle. On entend par solution globale celle qui englobe tous les miroirs de tous les sources dans tout le web. Une solution de telle envergure se heurte à des problèmes de distribution pour le stoquage, des problèmes de recherche et de mise à jour de la base de données. La solution spécifiée ici est une solution non globale mais qui vise à être globale dans les besoins de chaque organisme. Les organismes qui ont des intérêts communs doivent s'intéresser aux mêmes miroirs. Une certaine localité dans la recherche des documents sur le web justifie le choix de la coopération non globale et plus ciblé.

Une solution non globale doit se définir des limites. Ces limites portent particulièrement sur le nombre d'organismes qui peuvent coopérer entre eux et sur la taille de la BDM supporté par chaque organisme. Ces limites vont dépendre de la quantité d'information à échanger entre les organismes, de la latence du réseau, des algorithmes de filtrage d'information, etc.

Les travaux de recherche futurs sont nombreux en ce qui concerne les solutions possibles. Dans ce rapport on a cité les plusieurs axes de recherche auxquels on ne doit pas dissocier complètement du problème de la découverte, publication, localisation et mise à jour des miroirs. Ces travaux doivent être menés en parallèle avec des mesures de performance dans le but d'associer la solution choisie avec les résultats obtenus.

8 Remerciements

Je voudrais remercier Mesaac Makpangou pour des nombreuses heures de discussion et son appui constant dans le choix du thème de cette recherche. Ce rapport a béneficié également de discussions et de critiques :] de Christian Khoury et Guillaume Pierre, auxquels je suis reconnaissante. J'aimerais exprimer un grand merci à Ian Piumarta pour les innombrables consultations de tous genres :) sans lesquelles ce rapport n'existerait pas.

Références

[APS98] Yair Amir, Alec Peterson, and David Shaw. Seamlessly selecting the best copy from iternet-wide replicated web servers. In *DISC'98 - 12th International Symposium on DIStributed Computing*, Andros, Greece, Sep 1998.

[BDM98] Vincent Bouthors, Olivier Dedieu, and Mesaac Makpangou. Recommendations sharing forum for internet communities. Url, presentation, Dyade, Bull, INRIA, 1998. `http://webtools.dyade.fr/pharos/`.

[BL95] T. Berners-Lee. Hypertext transfer protocol http/1.0. " ", HTTP working Group Internet draft, October 1995.

[Lit91] Marc C. Little. *Object replication in a Distributed System.* PhD thesis, University of Newcastle upon Tyne Computing Laboratory, Newcastle upon Tyne, England., September 1991.

[Moc87] Paul Mochapetris. Dns : The domain name system. RFC 1034, Network Working Group, ISI, November 1987.

[MPKD98] Mesaac Makpangou, Guillaume Pierre, Christian Khoury, and Neilze Dorta. Replicated directory service for weakly consistent replicated caches. Rapport de Recherche RR-3514, INRIA, April 1998. `http://www-sor.inria.fr/publi/RDSWCDC_rr3514.html`.

[PSTT95] Karin Petersen, Mike Spreitzer, Douglas Terry, and Marvin Theimer. Bayou: Replicated database services for world-wide applications. In *Proceedings of the 15th ACM Symposium on Operating Systems Principles (SOSP-15)*, pages 3–6, Copper Mountain Resort, Colorado, December 1995. ACM.

[SHTH98] Maarten van Steen, Philip Homburg, Andrew S. Tanenbaum, and Franz J. Hauck. Locating objects in wide-area systems. *Communications Magazine*, pages 2–7, jan 1998.

[VtYB97] P. Vixie, S. thomson, Y.Rekhter, and J. Bound. Dynamic updates in the domain name system (dns update). RFC 2136, Network Working Group, ISI, April 1997.