

Thibaut CARPENTIER

ECOLE CENTRALE DE LYON

27 août 2004

Rapport de Stage
Grilles de calcul pour Modalys

Lieu du stage : IRCAM



1, place Igor-Stravinsky
75 004 Paris

Tuteur de stage : François DÉCHELLE



REMERCIEMENTS

Je tiens à remercier le personnel de l'IRCAM et en particulier François DÉCHELLE et Patrice TISSERAND pour m'avoir accueilli durant ce stage, pour leurs nombreux conseils et enseignements, ainsi que pour leur patience et leur tolérance.

Grâce à eux j'ai pu agréablement me familiariser avec les logiciels libres.

Enfin je souhaite remercier Simon Schampijer pour sa compagnie durant mon stage.



Table des matières

1	Présentation de l'IRCAM	4
1.1	La Recherche	4
1.2	La Création	4
1.3	La Transmission des savoirs	4
2	Présentation de l'équipe LOGICIELS LIBRES ET INGÉNIERIE LOGICIELLE	5
2.1	Projets antérieurs	5
2.2	Concert virtuel réparti	6
2.3	Grilles de calcul	7
3	Programme du stage	8
4	Présentation de MODALYS	9
4.1	Un synthétiseur par modèles physiques	9
4.2	Principe général de fonctionnement	9
5	Présentation du GRID COMPUTING	10
5.1	Les ressources d'une grille	10
5.1.1	La puissance de calcul (<i>Computation</i>)	10
5.1.2	La capacité de stockage (<i>Storage</i>)	10
5.1.3	Les communications	11
5.1.4	Licences et software	11
5.2	Exemples de grid	11
6	BOINC et GLOBUS TOOLKIT	13
6.1	Présentation de BOINC	13
6.2	Présentation de GLOBUS TOOLKIT	13
7	Présentation d'XTREMWEB	15
7.1	Introduction	15
7.2	Principe	15
8	Travail effectué au cours du stage	17
8.1	Contexte	17
8.2	Worker	17
8.2.1	Script Modalys pour XtremWeb	17
8.2.2	Fichiers Modalys (.mos) et XtremWeb	18
8.2.3	Fichiers temporaires de Modalys	18



8.2.4	Upload des fichiers binaires vers workers Mac OS X	19
8.2.5	Lancement du worker XtremWeb au boot	20
8.3	Client	21
8.3.1	Gestion des droits du client	21
8.3.2	Script pour l'envoi de plusieurs jobs Modalys	21
8.3.3	Utilisation du Client via VPN	22
8.4	Interface Web	22
8.4.1	User <i>UID</i> absent lors de la création d'un utilisateur	22
8.4.2	Suppression des fichiers sur le serveur	22
8.4.3	Nombreuses petites imperfections sur le portail	23
8.5	Divers	23
8.5.1	Packaging pour Mac OS X	23
8.5.2	Tests de performance	23
8.6	Récapitulatif du travail effectué	25
9	Conclusion et Perspectives	26
10	Bibliographie	27
11	Annexes	28
11.1	Guide pour l'utilisation d'XtremWeb	28
11.2	Fichier build.conf	30
11.3	Script Modalys pour XtremWeb - Version Linux	32
11.4	Script Modalys pour XtremWeb - Version Mac OS X	33
11.5	Envoi des jobs	34
11.6	Réception des jobs	37
11.7	StartupItems pour Mac OS X	39



1 Présentation de l'IRCAM

Fondé en 1969 par Pierre Boulez, l'IRCAM¹ est une institution musicale associée au Centre Pompidou et dirigée depuis 2002 par Bernard Stiegler (qui succède à Laurent Bayle). Chercher, créer et transmettre sont les pôles autour desquels se développe l'activité de l'IRCAM qui réunit en un même lieu des scientifiques et des musiciens, afin de les inciter à explorer ensemble des voies artistiques innovatrices.

1.1 La Recherche

L'IRCAM mène des recherches fondamentales sur les apports de l'informatique, de la physique et de l'acoustique à la problématique musicale. Elles ont pour vocation principale la mise au point d'outils logiciels qui viennent enrichir l'invention du compositeur et suscitent des échanges internationaux avec les grandes institutions universitaires ou de recherche. Des partenariats se développent également avec le monde industriel (réseaux, téléphonie, automobile...).

1.2 La Création

L'IRCAM invite dans ses studios de nombreux compositeurs. Chaque année, 20 à 25 oeuvres sont réalisées, qui associent des interprètes classiques (instrumentistes et chanteurs) et des nouvelles techniques. Ces musiques sont ensuite présentées au public, à Paris et en tournées, dans le cadre de saisons musicales organisées conjointement avec l'Ensemble Intercontemporain. Elles sont également reprises au plan international par de nombreuses formations. De plus, chaque mois de juin, l'IRCAM organise son propre festival, Agora², qui associe la création musicale à d'autres disciplines artistiques (danse, théâtre ou cinéma).

1.3 La Transmission des savoirs

L'IRCAM propose plusieurs programmes pédagogiques, notamment une formation doctorale, un cursus annuel d'informatique musicale destiné aux compositeurs et de nombreux ateliers, conférences ou débats à l'adresse de larges publics. Parallèlement, la médiathèque très largement informatisée met à la disposition des chercheurs, étudiants et mélomanes un important fonds musical. Enfin, l'IRCAM diffuse ses activités sous forme de livres et revues, de disques compacts et de CD-Rom.

¹Institut de Recherche et de Coordination Acoustique/Musique.
Site Internet : <http://www.ircam.fr>

²Pour plus de détails, voir <http://agora2004.ircam.fr>

2 Présentation de l'équipe LOGICIELS LIBRES ET INGÉNIE- RIE LOGICIELLE

L'équipe LLIL (Logiciels Libres et Ingénierie Logicielle³) a pour but de promouvoir la diffusion de logiciels libres issus de l'IRCAM et d'assurer une expertise technique sur les questions d'architecture logicielle.

Cette activité répond à un double objectif :

- la valorisation, soutenue au niveau national et européen, du développement collaboratif de logiciels selon le modèle libre⁴,
- une diffusion élargie des travaux de l'IRCAM auprès des communautés de la recherche et du secteur industriel.

L'équipe est constituée de deux membres permanents : François DÉCHELLE⁵, responsable de l'équipe (et tuteur du stage) et Patrice TISSERAND⁶, chargé de développement, auxquels viennent s'ajouter les stagiaires.



2.1 Projets antérieurs

Les projets précédents de l'équipe s'articulaient autour du développement des logiciels libres de l'IRCAM : jMax (un environnement de programmation visuelle pour l'interaction musicale temps réel), OpenMusic (un environnement de programmation visuelle pour la composition) et SDIF (une bibliothèque et des outils pour la manipulation de fichiers SDIF).

Enfin, le projet AGNULA (acronyme pour A GNU/Linux Audio distribution⁷) a pris fin en avril 2004. Ce projet coordonné par le Centre Tempo Reale à Florence⁸ impliquait d'importants centres de recherche et institutions : l'IRCAM, l' Universitat Pompeu Fabra, Music Technology

³<http://freesoftware.ircam.fr>

⁴L'expression "logiciel libre" fait référence à la liberté pour les utilisateurs d'exécuter, de copier, de distribuer, d'étudier, de modifier et d'améliorer le logiciel. Pour plus d'informations, voir notamment : <http://www.gnu.org/philosophy/philosophy.fr.html#AboutFreeSoftware>

⁵Francois.Dechelle@ircam.fr

⁶Patrice.Tisserand@ircam.fr

⁷<http://www.agnula.org>

⁸<http://www.centrotemporeale.it>

Group de Barcelona, Kungl Tekniska Högskolan (KTH), Music Acoustics Group de Stockholm, la Free Software Foundation Europe ⁹ et Red Hat France ¹⁰.

La tâche principale du projet AGNULA fut le développement de deux distributions de référence du système GNU/Linux complètement basées sur des logiciels libres (c'est-à-dire sous des licences libres approuvées par la FSF) et dédiées au développement d'applications audio et multimédia professionnelles et grand public. Une des distributions est basée sur une distribution Debian (DeMuDi) alors que l'autre distribution est basée sur une distribution RedHat (ReHMuDi). Ces deux distributions sont disponibles au téléchargement sur Internet ¹¹ et sur CD.

Les projets en cours sont centrés sur les architectures distribuées, plus particulièrement les applications multimédia distribuées sur Internet et les grilles de calcul.

2.2 Concert virtuel réparti

Un premier projet, le *concert virtuel réparti*, vise à répartir chaque acteur d'un concert (les musiciens, l'ingénieur du son ainsi que le public) sur Internet. Toutes les personnes participant à cette performance sont géographiquement séparées et communiquent par streaming audio temps réel sur Internet.

Les technologies développées dans ce projet (streaming audio temps réel sur Internet, contrôle temps réel distribué, spatialisation) sont susceptibles d'applications dans des domaines autres que les applications musicales, en particulier les applications distribuées de réalité virtuelle ou augmentée.

Ce projet offre un cadre d'expérimentation original pour les nouvelles fonctionnalités offertes par les réseaux haut débit : multicast natif et qualité de service.

Dans le cadre de ce projet de concert virtuel réparti, Simon Schampijer a développé une application client/serveur pour JACK (Jack Audio Connection Kit¹²) basée sur le protocole RTP¹³.

⁹<http://fsfeurope.org>

¹⁰<http://www.redhat.fr/>

¹¹<http://www.agnula.org/download/>

¹²<http://jackit.sourceforge.net/>

¹³CVS Repository at : <http://cvs.sourceforge.net/viewcvs.py/jackrtp/jackrtp/>



2.3 Grilles de calcul

L'environnement de travail du compositeur est traditionnellement centré sur l'ordinateur personnel qui bien que aujourd'hui très puissant limite considérablement la puissance de calcul et de stockage disponible. De plus, cette approche est très limitative du point de vue des possibilités ouvertes aujourd'hui par les réseaux haut débit. Enfin, la disponibilité de puissances de calcul et de stockage importante est également primordiale pour les chercheurs travaillant à l'IRCAM.

Le projet grilles de calcul vise d'abord à expérimenter les technologies de *Grid Computing* appliquées à la musique et au multimédia et ensuite à développer ces technologies pour les adapter aux besoins spécifiques de la musique et des média interactifs, par exemple en termes de temps de réponse et de latence garantis.

3 Programme du stage

La complexité algorithmique du logiciel *Modalys* ne permet pas pour l'instant de le faire fonctionner en temps réel. Les utilisateurs sont donc conduits à faire fonctionner le logiciel en mode *batch*, c'est-à-dire en lançant la synthèse d'un son à partir d'un fichier de paramètres (souvent produit par un autre logiciel) et en écoutant le son calculé à la fin de l'exécution. Ce mode d'utilisation est peu confortable et ne permet pas d'effectuer des essais portant sur un grand nombre de valeurs de paramètres différentes.

Pour offrir aux compositeurs un contexte d'utilisation plus commode, l'équipe "Logiciels libres et ingénierie du logiciel" a développé une solution basée sur les technologies de grilles de calculs, qui permet à un utilisateur de lancer un grand nombre de synthèses de sons par *Modalys* en parallèle, les calculs étant effectués sur les machines du réseau local de l'IRCAM.

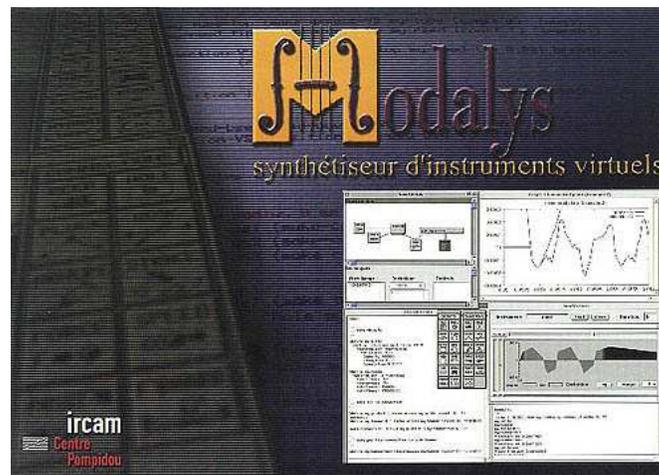
Programme du stage :

Le stage visera la finalisation de l'environnement d'exécution sur la grille de calcul locale et comprendra les phases suivantes :

- test de l'environnement : validation du fonctionnement en mode grille, évaluation de la commodité d'utilisation, étude des cas d'utilisation les plus fréquents,
- mesure de performance : comparaison de la puissance de calcul en mode grille par rapport au mode mono-machine,
- adaptation de l'application grille en fonction des résultats des tests et des mesures de performance,
- étude éventuelle de l'application des techniques de grille à d'autres applications.

Le stage se fera en collaboration avec l'équipe développant le logiciel *Modalys* et en fonction de leurs disponibilités avec un compositeur ou un assistant musical utilisant le logiciel *Modalys*.

4 Présentation de MODALYS



4.1 Un synthétiseur par modèles physiques

Modalys est un logiciel de synthèse sonore par modélisation physique qui traite un instrument de musique comme un assemblage de sous-structures élémentaires en interaction, chacune d'elle représentant un objet vibrant (plaque, corde, tube, membrane, etc...) dont le comportement peut être décrit par ses caractéristiques modales (linéaires), leurs couplages ayant un caractère non linéaire. A partir des "objets physiques", l'utilisateur construit son instrument, et décide du mode de jeu.

Modalys a été développé à l'IRCAM, initialement pour Macintosh.

4.2 Principe général de fonctionnement

Les étapes de la construction de l'instrument virtuel sont les suivantes :

- simulation des objets physiques constitutifs de l'instrument (corde, tube, etc) suivant les paramètres de l'utilisateur
- définition des accès aux objets (c'est-à-dire les emplacements où les modes de jeu seront appliqués)
- définition des connections qui indiquent les modes de jeu à pratiquer
- ajout de contrôleurs qui permettent de modifier les paramètres temporels d'un mode de jeu (et donc de "jouer" de l'instrument).

Tous ces paramètres peuvent être enregistrés dans un fichier de données qui pourra ensuite être lu par *Modalys*.

5 Présentation du GRID COMPUTING

Le *grid computing*, également appelé grilles de calcul, calcul partagé ou encore calcul distribué, consiste à exploiter pleinement les ressources de l'intégralité d'un parc informatique. L'objectif du grid computing, par analogie avec le réseau électrique (*power grid*), est de fournir à l'utilisateur les ressources dont il a besoin, sans se préoccuper de savoir où et par qui est produite l'énergie.

Cette idée de mutualisation des ressources informatiques est née d'une part de l'augmentation démesurée des besoins en puissance de calcul et de traitement des données dans le milieu de la recherche scientifique, et d'autre part du constat de la sous-utilisation générale des parcs informatiques. En effet les entreprises n'utilisent en général que 5 à 20 % des capacités de leurs ordinateurs pendant les heures ouvrables et pratiquement 0 % le reste du temps.

5.1 Les ressources d'une grille

Une grille est un ensemble de machines (serveurs ou postes de travail), parfois appelées "noeuds", "membres", "clients", "hosts", etc. Ces machines peuvent contribuer aux ressources de la grille de diverses manières :

5.1.1 La puissance de calcul (*Computation*)

La ressource la plus intéressante dans les grilles de calcul est la puissance de calcul fournie par les processeurs des noeuds. Il y a principalement trois méthodes pour exploiter les ressources calculatoires de la grille :

- La première est de lancer, sur la grille, une application existante sur une des machines, plutôt que de l'exécuter localement,
- la seconde est d'utiliser une application dont on peut séparer le travail en plusieurs parties qui peuvent être effectuées en parallèle sur différents processeurs,
- enfin la troisième est de lancer une application qui a besoin d'être exécutée un grand nombre de fois.

Pour évaluer la capacité d'une grille à utiliser les différents processeurs, on utilise le terme de *scalability*. Par exemple, si une grille à deux ordinateurs permet de réaliser un travail en deux fois moins de temps qu'un ordinateur, on dira que la *scalability* est parfaite.

5.1.2 La capacité de stockage (*Storage*)

Une des ressources les plus communément partagée sur les grilles est la capacité de stockage de données. On parle alors de *data grid*. La capacité de stockage est accrue en créant un système de fichiers qui permet d'utiliser l'espace libre sur les différents noeuds du réseau.

5.1.3 Les communications

Une autre ressource importante des grilles est la capacité de communication et d'échange de données. Ceci comprend les communications internes à la grille, mais aussi externes. Les communications au sein de la grille sont importantes pour soumettre les *jobs* et transmettre les données nécessaires à leur exécution.

5.1.4 Licences et software

On peut trouver dans une grille des softwares trop chers pour être installés sur chacune des machines de la grille. On peut alors utiliser la grille pour exécuter ces softwares sur les machines particulières sur lesquelles ils sont installés. Certaines licences de logiciels autorisent l'installation sur toutes les machines d'une grille, mais limitent le nombre d'exécutions simultanées. Les planificateurs (*schedulers*) des grilles peuvent généralement être configurés pour tenir compte de cela.

On retiendra qu'en général, et dans le cas particulier qui nous intéresse, les grilles sont principalement déployées dans le but de partager la puissance de calcul.

5.2 Exemples de grid

Les premières expériences de global computing avaient pour but de craquer des clés de cryptage, des codes RSA ou de trouver des nombres premiers.

A l'heure actuelle, la plupart des projets de grid demeurent des projets scientifiques. Ces réseaux de calcul distribué ne constituent pas des grilles à proprement parlé, mais plutôt des techniques de High Throughput Computing (HTC) consistant à exécuter la même application avec un grand nombre de paramètres différents. Ils illustrent néanmoins par leur parenté technique les bénéfices que l'on peut attendre du grid computing.

On peut ainsi mentionner quelques projets qui ont fait peu à peu connaître le grid computing au grand public :

- Seti@home¹⁴ : Ce projet américain, lancé en 1999, qui est toujours en cours, vise à rechercher la présence d'intelligences extraterrestres grâce au concours de plus de 5 millions de PC qui traquent des répétitions de signaux envoyés de l'espace captés depuis le radio-télescope d'Arecibo à Porto Rico. Avec une moyenne d'utilisation des stations d'environ 12 heures par jour, le projet a pu recueillir, depuis sa création, plus de 2 millions d'années de temps CPU, et de réaliser plus de $5.4e21$ opérations à virgule flottante.
- Le Decrypton¹⁵ : Fruit du travail de l'Association Française contre les Myopathies, d'IBM,

¹⁴<http://setiathome.ssl.berkeley.edu/>

¹⁵<http://www.genomining.com/home.fr.html>

de Platform Computing et de Genomining, ce projet HTC a permis la comparaison de plus de 550.000 protéines en deux mois grâce à la connexion de 100.000 PC de particuliers qui ont contribué chacun à 10 heures de calcul. Il aurait fallu environ 1200 années à un seul ordinateur pour effectuer les mêmes opérations,

- GIMPS¹⁶, The Great Internet Mersenne Prime Search, un programme de recherche de nombres de premiers de Mersenne¹⁷, lancé en 1997, qui a permis à ce jour de découvrir sept nouveaux nombres premiers, grâce au concours d'environ 70000 volontaires,
- Plusieurs autres projets peuvent être entrevus sur distributed.net¹⁸, site fondé en 1997

Les principaux critères que doivent vérifier les systèmes de Global Computing sont :

- extensibilité : les systèmes doivent être capables de gérer un grand nombre de machines (de plusieurs centaines à quelques millions),
- hétérogénéité : les workers doivent pouvoir fonctionner sous différents systèmes d'exploitation (et sur différents matériels),
- dynamicité : le nombre des workers (voire serveurs) doit pouvoir évoluer en permanence,
- disponibilité : le propriétaire d'une ressource doit pouvoir définir une politique de partage de cette ressource,
- tolérance aux pannes : le système doit supporter l'existence d'éléments défectueux. Pour cela, la plupart des grilles de computation effectue les calculs de façon redondante,
- utilisabilité : le système doit rester facilement programmable et maintenable,
- sécurité : un worker malicieux ne doit pas pouvoir corrompre l'application, un agent externe ne doit pas pouvoir se faire passer pour un serveur, etc.

Il existe trois principaux middlewares libres de grid computing. En voici une brève description.

¹⁶<http://www.mersenne.org/prime.htm>

¹⁷de la forme $2^p - 1$

¹⁸<http://distributed.net/projects.php.fr>

6 BOINC et GLOBUS TOOLKIT

6.1 Présentation de BOINC



BOINC, pour Berkeley Open Infrastructure for Network Computing¹⁹, est utilisé pour plusieurs projets de recherche scientifique de grande envergure, notamment SETI@HOME et Predictor@home²⁰. Ces projets sont indépendants, opérant chacun avec leur propre serveur et base de données.

Le client BOINC est actuellement disponible pour Mac OS X, Windows, Linux et d'autres systèmes Unix.

Cependant BOINC est plus orienté vers des projets de grande envergure à travers l'Internet (tel que Seti@Home). Il développe donc des outils qui ne sont pas indispensables dans le cas de grid computing pour les applications multimédia : des dispositifs de sécurité chargés d'éviter le piratage des crédits, des serveurs assurant des calculs redondants, etc.

6.2 Présentation de GLOBUS TOOLKIT



Débuté en 1996, le projet Globus est mené par Argonne²¹, ISI et University of Chicago. De nombreux projets s'appuient déjà sur le Toolkit Globus, dont le Grid Physics Network (GridPhyN), le Large Hadron Collider au CERN ou encore le Network for Earthquake Engineering and Simulation (NEES).

¹⁹<http://boinc.berkeley.edu/>

²⁰<http://predictor.scripps.edu>

²¹<http://www.anl.gov/>



Globus Toolkit est une implémentation open source de OGS (Open Grid Services Infrastructure). OGS est une spécification formelle et technique des concepts d'OGSA. OGSA (Open Grid Services Architecture) représente une nouvelle vision à la fois des *webservices* et des *gridservices*. En définissant des formats et des protocoles standards pour les communications, OGSA constitue un moyen de construire un grid-system interopérable et à grande échelle.

Bien qu'il tende à s'imposer comme le standard actuel parmi les solutions libres de grid computing, Globus Toolkit présente certaines limitations :

- son déploiement est difficile du fait de la durée de compilation : plus d'une heure sur une machine dotée de 512 Mo de RAM avec un processeur à 2,4 GHz
- son installation requiert près de 300 Mo
- il présente des bugs
- si la mise en place d'un cluster est facile, il n'en est pas de même pour un réseau hétérogène (avec des OS différents)

7 Présentation d'XTREMWEB



7.1 Introduction

*XtremWeb*²² est un projet de recherche dans le domaine des systèmes de grid computing, développé par l'Equipe Architectures Parallèles (sous-groupe "Cluster and Grid") du LRI²³. Il s'agit d'un software à but non lucratif, gratuit, open source et multiplateforme.

Comme les autres plateformes de systèmes distribués, XtremWeb utilise les ressources de stations (PC, postes de travail ou serveurs) connectées à l'Internet ou à travers un LAN. Parmi les utilisateurs ou collaborateur d'XtremWeb, on peut citer C@sper (projet de recherche industrielle entre Airbus et Alcatel), CGP2P ACI GRID (Projet français de recherche académique sur les Grilles de Calcul), EADS, et de nombreuses universités (Université de Lille, E.N.S. de Lyon). XtremWeb est basé sur le principe de vol de cycle (*cycle stealing*). D'un point de vue architecture, XtremWeb est composé de clients, de workers et d'un (ou plusieurs) serveurs.

7.2 Principe

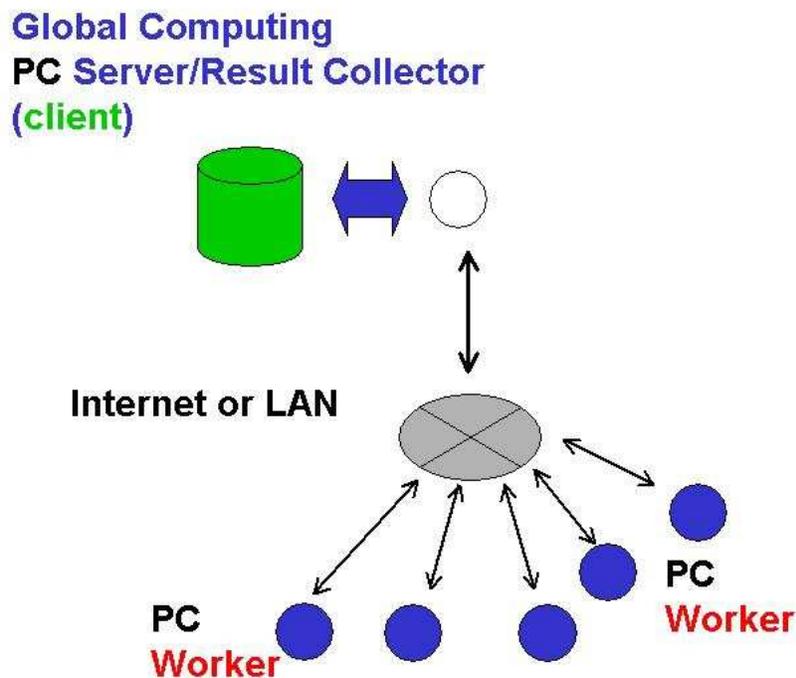
Le principe de fonctionnement est assez classique : un client soumet au serveur un job à effectuer. Les workers contactent régulièrement le serveur pour recevoir des jobs. En réponse, le serveur envoie un ensemble de paramètres, et éventuellement l'application à lancer, si celle-ci n'est pas déjà présente sur le worker. Quand le worker a terminé son job, il contacte le *collecteur de résultats* (c'est-à-dire le serveur), et envoie ses résultats. En fonction de la taille du résultat, la communication entre le serveur et les workers peut s'établir suivant différents protocoles. Avec la version actuelle, le worker peut envoyer des résultats d'une taille maximale de 100 MB. Enfin, le client recontacte le serveur pour recevoir les résultats des jobs qu'il a soumis.

²²<http://www.xtremweb.net>

²³Laboratoire de Recherche en Informatique, Université de Paris XI
<http://www.lri.fr>

XtremWeb peut aussi être utilisé pour construire un système Peer-to-Peer centralisé. En effet, chaque worker peut aussi être utilisé en tant que client, pour soumettre des jobs. Il est prévu, pour les versions futures, de permettre l'envoi des résultats des workers directement aux clients (sans passer par le serveur central), afin de limiter le besoin en bande passante et d'accélérer les communications entre les instances de la grille.

Architecture d'XtremWeb : Client(s) - Serveur(s) - Worker(s)



8 Travail effectué au cours du stage

8.1 Contexte

J'ai tout d'abord effectué un travail de documentation sur le grid computing en général, et sur les solutions libres existants dans ce domaine (XtremWeb et Globus Toolkit en particulier). Pour cela, je me suis, entre autres, appuyé sur le travail d'un précédent stagiaire (Julien Delacroix : *Mise en place d'une grille de calcul*, voir Bibliographie).

Au début du stage, une version d'XtremWeb (1.3.0) était déjà installée sur les machines de l'équipe, avec le serveur sur *lipari.ircam.fr*, et un worker sur chacune des 6 machines de l'équipe. La première partie du travail a consisté à installer une version plus récente, la 1.3.2, dont la date de release est le 10 juin 2004²⁴.

Pour compiler XtremWeb, il faut tout d'abord remplir le fichier *build.conf* (voir en Annexe) qui contient notamment les données relatives à l'administrateur du système XtremWeb, à la base de donnée, aux répertoires d'installation, etc. Pour installer le serveur XtremWeb, il est nécessaire d'avoir Java, MySQL et un serveur Apache, ce qui est le cas. La compilation peut alors avoir lieu. A l'issue de cette installation, on dispose d'un serveur, d'un client, d'un worker, de la base de données XtremWeb, et d'un portail web permettant d'effectuer certaines opérations de base (tels que l'envoi de jobs et la réception des résultats).

Après une période de découverte et de prise en main du logiciel (à la fois en utilisant l'interface web et en ligne de commande), mon travail a consisté à corriger les petits défauts/bugs constatés au fur et à mesure de l'utilisation, et à adapter la grille pour permettre l'exécution de job Modalys.

Dans les paragraphes qui suivent, je présente sommairement quelques corrections qui ont dû être apportées au middleware.

8.2 Worker

8.2.1 Script Modalys pour XtremWeb

Lorsqu'on ajoute une application à XtremWeb, le fichier binaire de cette application est copié par le serveur. Ensuite lorsqu'on lance un job nécessitant cette application, le fichier binaire est transféré au worker²⁵, qui l'exécute²⁶. Ceci présuppose que le fichier binaire se suffit à lui-même, c'est-à-dire qu'il ne fait pas appel à d'autres fichiers. Or ça n'est pas le cas de

²⁴La version 1.3.3 a été publiée le 17 juin 2004 puis la 1.3.4 le 21 juillet 2004 et enfin la 1.3.5 le 11 août 2004, mais il a été décidé de continuer à travailler avec la 1.3.2, dans un premier temps

²⁵dans `/tmp/XW.{worker.name}.{dispatcher.name}.cache/bin/`

²⁶dans `/tmp/XW.{worker.name}.{dispatcher.name}/works/{Job_UID}_{Job_Number}/scratch/`

Modalys, qui pour fonctionner, a besoin au minimum (si on ne tient pas compte de tous les plug-ins nécessaires notamment pour la lecture des fichiers sons générés) des fichiers contenus dans `Modalys/application/mos/`.

Si on ajoute simplement l'application Modalys à XtremWeb, ces fichiers ne seront pas transférés au serveur (ni aux workers), et donc une erreur surviendra lors du lancement du binaire Modalys.

Il faut donc envisager une autre stratégie. Une première solution envisageable aurait été de regrouper le binaire de Modalys et les fichiers de configuration nécessaire en un même exécutable. Cette solution présente un net inconvénient lié au poids de l'exécutable : le binaire de Modalys et ses fichiers de configuration pèse au minimum 13 Mo. Le transfert de ce fichier du serveur vers les workers serait donc lent, et encombrerait le réseau.

L'alternative retenue consiste à utiliser un script shell qui va appeler une version de Modalys présente sur la machine du worker. Avec cette solution, le fichier circulant (le script) pèsera tout au plus quelques Ko. Bien entendu ceci présuppose que chaque noeud de la grille dispose d'une version de Modalys installée, ce qui n'est pas le cas actuellement. De plus, le worker a besoin de savoir où se trouve l'application Modalys. Pour cela on crée un fichier de configuration (`modalys.conf`) dans lequel on indique l'emplacement de Modalys.

8.2.2 Fichiers Modalys (.mos) et XtremWeb

Les fichiers contenant les instructions pour Modalys sont les fichiers `.mos`. Ils doivent être transmis au worker. XtremWeb inclut deux méthodes pour passer des fichiers : soit en tant que `input files` (pour les applications qui ont besoin d'une entrée standard) soit en tant que `zipfiles` (pour les applications qui ont besoin d'une structure de répertoire dédiée). En raison de difficultés pour gérer la `stdin` de Modalys, nous sommes contraints de passer les fichiers `.mos` en tant que `zipfiles` (bien qu'il paraît plus logique de les passer comme `input files`), ce qui signifie que l'utilisateur doit zipper le(s) fichier(s) qu'il souhaite traiter, avant de lancer la tâche.

8.2.3 Fichiers temporaires de Modalys

Pour fonctionner, Modalys utilise des fichiers temporaires pour stocker des données de calculs. Il s'agit des fichiers `mosout.float`, `mosout.aiff` et `mosout.plot`. Par défaut ces fichiers sont stockés dans `/tmp/`, qui est un répertoire commun à tous les utilisateurs unix d'une même machine. Ceci implique notamment que plusieurs instances de Modalys ne peuvent être lancées simultanément.

Or un des impératifs du grid computing est de rendre le worker le plus transparent possible : l'utilisateur ne doit pas être perturbé par la présence d'un worker sur son poste. Notamment un worker qui serait en train d'exécuter une tâche Modalys ne doit pas empêcher l'utilisation

de Modalys par le propriétaire de la machine. Il est donc impératif de trouver une méthode permettant l'utilisation de plusieurs Modalys en même temps.

Nous choisissons pour cela d'écrire les fichiers temporaires (mosout) dans des répertoires différents pour chaque instance de Modalys. Pour cela il faut attribuer un nom unique au répertoire créé.

Une solution simple consiste à nommer le répertoire avec le numéro PID (Process ID : numéro d'identification unique) de l'instance Modalys. Les fichiers mosout sont créés, au démarrage de Modalys par le fichier d'initialisation /application/mos/init/elk/toplevel.scm²⁷. Il existe des commandes Scheme permettant de créer un répertoire ou de récupérer le PID d'une application. Ces commandes sont fournies par la librairie unix.o (disponible dans l'extension unix pour Elk) qui n'est pas fournie initialement avec Modalys.

Une première solution a donc consisté à ajouter cette librairie et à modifier le fichier toplevel.scm (en utilisant les fonctions unix-getpids, unix-mkdir, etc). La suppression des fichiers temporaires est assurée par une surcharge de la méthode "exit" (qui quitte l'interpréteur et Modalys) dans le fichier toplevel.scm

Cependant, ceci n'est pas conforme à notre souhait de modifier le moins possible Modalys. Ainsi nous allons en fait utiliser un script shell pour créer un répertoire unique, copier le fichier toplevel.scm existant et y modifier l'emplacement de destination des fichiers mosout. De cette manière, la version de Modalys installé localement demeure inchangée. Seul le worker va travailler avec des fichiers modifiés temporaires. Voir les scripts en annexe (un script pour la version Linux et sa variante pour Mac OS X).

En sus des fichiers mosout, Modalys génère également les fichiers temporaires /tmp/ld ??? (uniquement dans la version Linux). Après avoir quitter Modalys, ces fichiers sont vides, mais demeurent sur le disque. Leur destruction est également prise en charge par le script Modalys pour XtremWeb.

8.2.4 Upload des fichiers binaires vers workers Mac OS X

Lorsqu'une tâche est soumise par un client, le dispatcher vérifie que les workers disposent dans leur cache du fichier binaire correspondant à cette tâche. Si ce n'est pas le cas, il le transmet. Or dans le cas de worker tournant sous Mac OS X, les fichiers binaires, une fois reçus, ne présentent pas les droits en exécution. De fait, les jobs ne sont pas effectués.

L'implémentation de la méthode "chmodpx" (chargée d'ajouter les droits en exécution) pour Mac OS X se trouve dans le fichier XWUtilDummy.java. Cette méthode a été corrigée, afin

²⁷plus précisément par la fonction : (mos :initialize-modalys "/tmp/" "mosout.float" "/tmp/" "mosout.aiff" "/tmp/" "mosout.plot")

d'assurer le bon fonctionnement du worker sur cette plateforme.

8.2.5 Lancement du worker XtremWeb au boot

Comme dans la plupart des entreprises, les machines de l'IRCAM sont allumées en permanence. Ainsi, la majeure partie du temps, les ordinateurs sont allumés mais aucune session utilisateur n'est ouverte. Dans le cadre du grid computing, il est indispensable de pouvoir utiliser le cpu pendant ces périodes de très faible activité. Il faut donc que le worker puisse être lancé dès le démarrage du système, et qu'il soit opérationnel même si aucune session utilisateur n'est ouverte. En d'autres termes, il faut que le worker se comporte comme un démon (application qui tourne en permanence en arrière plan).

Sous Mac OS X (à partir de la version 10.3), il existe deux méthodes équivalentes pour lancer un démon au boot : on peut soit ajouter un *StartupItem* soit utiliser les *rc scripts*.

Les StartupItems sont des applications lancées au boot ; à chaque application correspond un dossier soit dans /System/Library/StartupItems, soit dans /Library/StartupItems. Ce dossier doit contenir un script (chargé d'appeler l'application ou le démon) et un fichier de configuration (.plist) qui indique les dépendances entre les démons et leur ordre de lancement.

Les StartupItems sont un mécanisme conçu pour les versions de Mac OS X antérieures à 10.3, mais maintenu jusqu'à présent.

A partir de 10.3, Apple recommande l'utilisation du script rc pour lancer un démon. Cette nouvelle approche consiste à enregistrer le démon dans le fichier /etc/mach_init.d (pour un lancement indépendant des interfaces utilisateurs) ou /etc/mach_init_per_user.d (pour un lancement lors de l'ouverture de session).

Dans un souci de compatibilité avec les différentes versions de Mac OS X que l'on peut rencontrer dans le parc IRCAM, il a été choisi d'utiliser les StartupItems (voir Annexe : StartupItems pour Mac OS X) pour le lancement du worker. Néanmoins une évolution vers les rc scripts est à prévoir.

En essayant de lancer le worker au boot, nous avons découvert un nouveau problème avec XtremWeb : celui-ci ne se comportait pas comme un démon mais comme une application, c'est-à-dire qu'il utilise une interface graphique. De fait, lorsqu'un utilisateur quitte sa session, il est contraint à stopper le worker.

Il est alors apparu que ce comportement est dû à des appels à des classes graphiques (java.awt.*) dans la méthode Debug. Ces appels ont pour but d'associer une couleur spécifique à chaque type de message de debug (info, warning, error, etc.). Ceci étant inutile, et fort gênant, ça a été supprimé, afin qu'il n'y ait plus de dépendance sur awt (du moins pour le worker. On aurait pu faire de même pour le client, mais on considère qu'il ne sera utilisé qu'à l'intérieur d'une session.).

Enfin, il convient de signaler que le worker est lancé avec une basse priorité (voir Fichier XWorker dans StartupItems, en Annexe), afin de ne pas utiliser de ressources si l'utilisateur en a besoin. En lançant le worker à basse priorité, tous les processus fils appelés par le worker (en d'autres termes Modalys appelé par le worker) sont lancés à basse priorité.

8.3 Client

8.3.1 Gestion des droits du client

Plusieurs bugs ont été mis en évidence et signalés aux développeurs d'XtremWeb lorsqu'ils relevaient d'un problème de fond. Notamment un problème critique a été détecté dans la gestion des droits des utilisateurs XtremWeb. XtremWeb instaure différents niveaux de permission pour les clients, définis en fonction du paramètre entier "right" de l'utilisateur :

- un utilisateur dont le right est fixé à 0 n'a aucun droit
- $\text{right} \geq 1$ permet de lister les jobs du client
- $\text{right} \geq 2$ permet de lister les applications
- $\text{right} \geq 3$ permet de récupérer des jobs
- $\text{right} \geq 4$ permet d'insérer des jobs
- $\text{right} \geq 5$ permet de supprimer des jobs (un utilisateur "normal" a ses droits fixés à 5)
- $\text{right} \geq 6$ permet d'ajouter des applications
- $\text{right} \geq 7$ permet de supprimer des applications (le niveau 7 correspond aux "advanced users")
- $\text{right} \geq 8$ permet d'ajouter des users
- $\text{right} \geq 9$ permet de supprimer des users
- enfin $\text{right} = 100$ correspond au "super user" qui a tous les droits.

Lorsque le client souhaite effectuer une tâche, la méthode checkClient est appelée pour tester le right level de l'utilisateur et autoriser ou refuser la tâche. Cependant une erreur dans le code de la méthode checkClient a été observée, provoquant des viols de permissions.

Suite à nos remarques, les développeurs d'XtremWeb ont corrigé ce problème dans la version 1.3.5 (sortie le 10 août 2004). L'implémentation correcte de checkClient a donc été rajoutée dans la version 1.3.2 que nous avons décidé de conserver.

8.3.2 Script pour l'envoi de plusieurs jobs Modalys

Afin de faciliter l'utilisation du client XtremWeb en ligne de commande, un script shell a été conçu pour envoyer des jobs Modalys. La syntaxe pour l'emploi de ce script est simple :
sh submit_job modalys fichier_a_traiter.mos (il suffit que l'utilisateur passe en argument la liste des fichiers mos à traiter).

Le script client (Voir en Annexe : `submit_job`) pour XtremWeb effectue la compression des fichiers et prend en compte le cas de l'envoi de plusieurs fichiers mos. Ensuite le script client établit simultanément autant de requêtes qu'il y a de fichier mos.

Ainsi le dispatcher envoie les jobs aux différents workers de la grille, et la durée de traitement est (a priori) diminuée. Une fois les jobs terminés, le script client récupère les résultats (qui sont enregistrés dans autant de répertoire que de jobs). Enfin, un autre script (Voir en Annexe : `retriev_job`) est à lancer lorsque l'utilisateur souhaite récupérer les résultats.

8.3.3 Utilisation du Client via VPN

Un test d'utilisation du Client via tunnel VPN²⁸ a été réalisé et a prouvé la possibilité d'utiliser le client depuis "l'extérieur" de l'IRCAM : à partir d'une machine distante mais faisant partie du VPN de l'IRCAM, il est possible de faire fonctionner le worker ou le client.

Contrairement aux projets "de type Seti@Home", l'intérêt ne repose pas sur l'utilisation distante du worker, mais du client. En effet, la plupart des compositeurs utilisant Modalys ne sont pas résidents IRCAM, mais, moyennant une autorisation pour établir une connexion VPN, ils peuvent utiliser les ressources CPU du réseau local de l'IRCAM.

8.4 Interface Web

8.4.1 User UID absent lors de la création d'un utilisateur

Le portail web dispose d'une page (cachée) (`{install_dir}/www/private/register.php`) permettant de gérer les users xtremweb (création, édition, suppression d'un utilisateur xtremweb). Or, en créant un utilisateur à l'aide de cette page php, on constate, dans la base de donnée MySQL, que l'utilisateur crée ne dispose pas de User UID. Ce problème a été corrigé, le nouveau User UID valant : `{u_id} = md5($V_Login.confGetPassPhrase());`

Un problème analogue se présente lors de la création d'un user en ligne de commande²⁹ : l'utilisateur créé possède un UID mais celui-ci ne correspond pas à la valeur attendue³⁰. Ce problème n'a pour l'instant pas été traité, et l'interface web demeure une façon rapide et pratique d'ajouter des utilisateurs.

8.4.2 Suppression des fichiers sur le serveur

A partir de l'interface web, un utilisateur peut, après avoir downloader ses résultats, choisir de supprimer les fichiers résultats du serveur. Cependant cette opération ne supprime pas les

²⁸Virtual Private Network

²⁹`java -jar XWClient.jar -xwconfig xwclient.conf -xwadduser <login> <password> <email> <rights>`

³⁰`md5($V_Login.confGetPassPhrase())`

fichiers d'entrée (zipfiles ou input files) qui ont été envoyés par le client sur le serveur. Ceci provoque donc un encombrement inutile du disque de la machine serveur. La fonction "delete" (`{install_dir}/www/php/db/library.php`) du portail a donc été modifiée de sorte que les input files soient supprimés en même temps que les output files.

8.4.3 Nombreuses petites imperfections sur le portail

Le portail web présentait de nombreuses petites imperfections qui ont été corrigées lorsqu'elles ont été détectées. Notamment des liens erronés, un système d'envoi d'email (permettant à un nouvel utilisateur de faire une demande de registration ou de notifier un bug auprès du webmaster xtremweb) très peu convivial, etc.

8.5 Divers

8.5.1 Packaging pour Mac OS X

Afin d'assurer une expansion de la grille, il est indispensable de créer des packages faciles à installer pour Mac OS X.

Il existe déjà des packages pour Modalys. La version utilisée pour réaliser les essais est la 2.3.5.

Nous avons créé des packages Mac OS X permettant l'installation du worker, et celle du client. Ces packages sont disponibles au téléchargement sur le portail web interne XtremWeb à l'IRCAM. Ils pèsent chacun environ 3 Mo.

L'installation du worker copie les fichiers dans /Applications et ajoute XWWorker aux StartupItems. Ainsi le worker est mis en marche lors du prochain boot de la machine. Le fichier de configuration du worker est le même pour toutes les machines (nom d'utilisateur : worker-mac).

Le package pour le client copie l'archive jar du client, le script pour soumettre des jobs et le fichier de configuration dans /Applications. Ensuite l'utilisateur doit remplir le fichier de configuration avec son login et mot de passe (qu'il peut obtenir sur le portail, dans la section Download Client), afin de pouvoir utiliser le client.

8.5.2 Tests de performance

Les grilles de calcul sont a priori prévues pour des calculs extrêmement lourds (plusieurs heures). Or ça n'est pas le cas de Modalys : les fichiers à traiter correspondent en général à des sons de quelques secondes, et leur calcul dure quelques minutes. De fait, le temps dû aux communications entre les entités de la grille n'est plus négligeable, et les performances sont loin d'être optimales. La conception actuelle de la grille pour Modalys se révèle surtout efficace

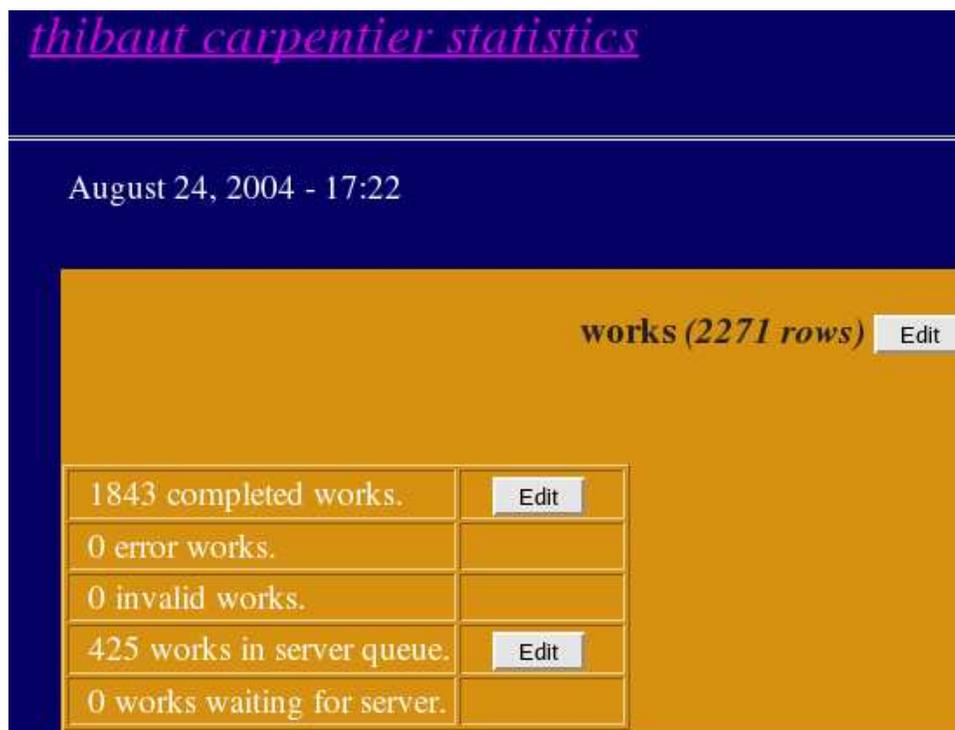
(d'un point de vue de l'utilisateur) dans le cas d'un grand nombre de fichiers Modalys à calculer.

Afin de s'assurer de la stabilité du système, et d'évaluer la gêne occasionnée sur les postes munis de worker, nous avons dans un premier temps mené des tests sur un échantillon restreint de worker. Durant cette période d'évaluation, un grand nombre de jobs modalys lourds (correspondant à des échantillons sonores de plusieurs minutes) a été lancé (plus de 1000 jobs par jour, dispatchés sur 5 workers actifs).

A la suite de ces tests, il semble que l'activité du worker (et surtout des applications tournées par le worker) ne perturbe pas l'utilisation "normale" d'une station de travail.

Le déploiement à plus grande échelle de la grille (c'est-à-dire des workers) au sein du réseau IRCAM est donc prévue. La diffusion de l'information reste à faire particulièrement du côté des compositeurs et utilisateurs de Modalys, puisque ce sont eux les bénéficiaires de cette opération.

Les statistiques de la grille





8.6 Récapitulatif du travail effectué

En définitive, mon travail a consisté à écrire un script permettant de lancer Modalys sur une machine-worker, à faire tourner ce worker en permanence et de façon transparente, sur les deux plateformes majoritairement présentes à l'IRCAM : Linux et Mac OS X.

De plus, un des objectifs principaux a été de rendre l'installation et l'utilisation de la grille (du worker comme du client) la plus "user-friendly" possible. Pour cela il a été nécessaire d'écrire des scripts simples et rapides d'utilisation, ainsi que de créer des packages facilitant l'installation.

Un patch contenant l'ensemble des modifications (environ 600 lignes) apportées au code source est disponible³¹. Il s'agit d'un *diff* entre les sources originelles de la 1.3.2 et la version utilisée.

De même un patch contenant les modifications apportées au portail web (environ 500 lignes) est créé³².

³¹dans /data/xtremweb/patches/src.patch

³²dans /data/xtremweb/patches/www.patch

9 Conclusion et Perspectives

Ce stage m'aura permis d'appréhender de façon pratique et concrète le concept de grille de calcul qui, à n'en pas douter, est amené à se développer de façon importante dans les domaines industriels et de la recherche scientifique.

De plus, ces trois mois ont été l'occasion de découvrir et d'adopter les logiciels libres, et de me familiariser ou d'approfondir certaines notions (langage bash, java, bases de données MySQL, etc.).

La mise en place d'une grille de calcul opérationnelle au sein du réseau local de l'IRCAM ouvre de nouvelles perspectives. C'est pourquoi il pourra être intéressant d'envisager :

- Une adaptation plus précise de la grille aux besoins des compositeurs utilisant Modalys : en général les compositeurs synthétisent note après note ou son après son, il faudrait alors lancer chaque note comme un job sur la grille puis automatiser la collecte et l'“assemblage” des différents sons obtenus. On peut également envisager une prise en compte des éventuelles dépendances entre les fichiers mos (certains fichiers peuvent avoir besoin des résultats issus d'autres fichiers),
- La parallélisation du code de Modalys : Il pourrait être utile de paralléliser l'algorithme de calcul de Modalys, c'est-à-dire de dégager plusieurs sous-algorithmes non corrélés qui seraient lancés comme des jobs indépendants sur la grille. Néanmoins, la parallélisation est utile dans le cas de calculs complexes et fastidieux. Dans le cas particulier de “petits” fichiers Modalys à traiter, elle semble largement superflue,
- L'utilisation de la grille pour d'autres applications : La grille étant mise en place, il est assez facile d'ajouter de nouvelles applications, afin d'augmenter son utilisation et donc d'optimiser l'utilisation des ressources. Notamment dans le cas de développement de modèles physiques, il peut être intéressant de tester de grands nombres de paramètres à travers la grille.

10 Bibliographie

1. *XtremWeb User Guide (version 1.3.0)*, Gilles Fedak, Oleg Lodygensky & Tangui Morlier, 2003, ©by Laboratoire de Recherche en Informatique
<http://www.lri.fr/fedak/XtremWeb/doc/xtremweb.pdf>
2. *Introduction to Grid Computing with Globus*, Luis Ferreira, 2003, IBM RedBooks
<http://www.redbooks.ibm.com/redbooks/pdfs/sg246895.pdf>
3. *Fundamentals of Grid Computing*, Luis Ferreira, 2002, IBM RedBooks
<http://www.redbooks.ibm.com/redpapers/pdfs/redp3613.pdf>
4. *Rapport de stage TN09 : Mise en place d'une grille de calcul*, Julien Delacroix, UTC, 2004
5. *Documentation Apple pour développeurs : Le Processus de démarrage du système*
<http://developer.apple.com/documentation/MacOSX/Conceptual/BPSystemStartup/>
6. *Bachelor Thesis - Virtual Distributed Concerts*, Simon Schampijer, 2004
7. *Public Computing : Reconnecting People to Science*, Dr. David P. Anderson, Presented at the Conference on Shared Knowledge and the Web, Residencia de Estudiantes, Madrid, Spain, Nov. 17-19 2003.
http://boinc.berkeley.edu/talks/madrid_03/madrid.html

11 Annexes

11.1 Guide pour l'utilisation d'XtremWeb

Mémo pour XtremWeb

Version d'XtremWeb utilisée : 1.3.2

Host : lipari (pour le serveur XW et la base de donnée MySQL)

Répertoire d'installation : /data/xtremweb/XW132/install et /data/xtremweb/XW132/www

Sources : /data/xtremweb/XW132/builddir

Base de donnée utilisée : xtremweb_1_3_2

Administrateur de la grille : xwadmin (right = 100)

Login des workers Mac Os X : worker-mac (right = 5)

Login par défaut des clients : anonymous (password = anonymous) (right = 5)

Portail web : <http://lipari.ircam.fr/xtremweb>

Ajout d'utilisateurs : <http://lipari.ircam.fr/xtremweb/private>

Version de Modalys utilisée : 2.3.5

Répertoire du client : /data/xtremweb/Client

Scripts : /data/xtremweb/scripts

Patches des modifications apportées à la version 1.3.2 : /data/xtremweb/patches

Descriptif des scripts :

- modalyssscript-Linux.sh : script pour l'application Modalys pour le worker Linux
- modalyssscript-Mac OS X.sh : script pour l'application Modalys pour le worker Mac OS X
- install-XWWorker-1.3.1-MacOSX.sh : script permettant d'installer un worker sur Mac OS X (mais n'ajoute pas de StartupItem)
- install-XWWorker-Linux.sh : script permettant d'installer un worker sur Linux
- submit_job : script pour envoyer des jobs (aide : submit_job -h)
- retrieve_job : script pour récupérer les jobs lancés (dont le JobID est stocké dans /tmp/JobID.txt)
- submit_and_retrieve_job : script qui envoie les jobs est reste en attente jusqu'à ce qu'ils soient terminés, puis récupère les résultats

Mémo commandes

Lancement du serveur :

```
/data/xtremweb/XW132/install/bin/rc.xtremweb.server start
```

(respectivement stop | restart | status)

Utilisation du client :

```
java -jar XWClient-Version.jar -xwconfig xwclient.conf -xwhelp
```

où Version = Mac-OSX ou Linux

Envoi / Récupération de jobs :



```
./submit_job modalys file1.mos file2.mos etc  
Puis pour récupérer les derniers jobs lancés :  
./retriev_job
```

Astuce

Lorsqu'on ajoute une application à la grille de calcul, il faut indiquer la plateforme et le type de cpu pour cette application. Il est alors affecté un UID à cette application. Or, dans le cas d'applications multi-plateformes, il peut être utile qu'elles aient le même UID.

Par exemple, on souhaite lancer l'application Modalys à la fois sur Mac OS X et sur Linux. Lorsqu'on ajoute l'application Modalys pour Linux, un UID lui est affecté, et le binaire correspondant est placé dans :

```
/data/xtremweb/XW132/www/db/{UID}/bin/{UID}-Linux-ix86
```

Pour que l'application Modalys Mac OS X porte le même UID, il suffit de placer le binaire pour Mac OS X dans le même répertoire et de le nommer {UID}-Mac OS X-ppc.

Ainsi lorsqu'on lance un job pour l'application désignée par "modalys", les jobs sont lancés indifféremment vers Linux ou Mac OS X.



11.2 Fichier build.conf

```
#####  
## XtremWeb building and installing configuration file ##  
#####  
  
# Definition of the administrator of your XtremWeb platform  
=====
```

xtremweb.admin.login=xwadmin
xtremweb.admin.password=xwadmin
xtremweb.admin.email=dechelle@ircam.fr
xtremweb.admin.fname=Ã§Franois
xtremweb.admin.lname=Ã©Dchelle

host running XtremWeb server
dispatcher.host=lipari.ircam.fr
dispatcher.rmiregistry.port=4322

Database system name (mysql is default) and its host
db.system=mysql
db.host=localhost

#Database information
=====

Database administrator and his password, if any
used to create xtremweb db user and database
db.su.login=root
db.su.password=
XtremWeb database name
db.name=xtremweb_1_3_2

XtremWeb Directories
=====

#The installation directory
install.dir=/data/xtremweb/XW132/install
#The directory where the xtremweb php interface will be installed
install.www.dir=/data/xtremweb/XW132/www

#unix xtremweb user
install.user=xtremweb
#unix apache user
install.www.user=apache

Security stuff
=====

pass sentence to generate keys/passwords
xw.passwordPass="some_chars_to_generate_keys"

MORE BUILD OPTIONS #
#####

Xtremweb Webmaster informations
These will be inserted in PHP pages so that user can contact you

By default, the administrator info defined below is used

webmaster.name=dechelle



```
webmaster.address=dechelle@ircam.fr

# XtremWeb db user (used by xtremweb server to communicate with the db
#
# By default, the administrator info defined bellow is used
#
db.user.login=xtremweb_1_3_2
db.user.password=xtremweb_1_3_2

# Uncomment the following properties to
# compile a windows version from linux
#target.os=Windows NT
# prefix of mingw if we are cross-compiling
#cross.prefix=/usr/bin/i386-mingw32msvc-
#cross.java.home=/work/JDK 14

#remote installation
#rinstall.host=irlin2
#rinstall.login=demo_ra
#rinstall.password=demo_ra
#rinstall.prefix=XtremWeb/slash/xtremweb

#option to process the idl files
xwidl.opts="-verbose_ -headers_ -interface_ -java_ -rmi_ -comm_ -handler_ -file_"

# SMTP host (this MUST be double quoted!)
smtp.host="localhost"

copysrc=on
```



11.3 Script Modalys pour XtremWeb - Version Linux

```
#!/bin/sh

# Reading Modalys binary file location from configuration file (/data/tcarpent/modalys.conf)
export bindir='cat $HOME/.modalys.conf | grep binary_location = | sed 's/binary_location=/'

# Creating temporary folder for initialization files
tmpfolder=/tmp/$$-modalystmp
mkdir $tmpfolder
cp -r $bindir/mos/ $tmpfolder

# Avoid "normal" initialization of temp files
toplevel_in=$bindir/mos/init/elk/toplevel.scm
toplevel_out=$tmpfolder/mos/init/elk/toplevel.scm
sed -e 's/(mos:initialize /;;;(mos:initialize /' $toplevel_in > $toplevel_out

# Initialize modalys in tmpfolder
init="(mos:initialize-modalys" \" $tmpfolder /\ " \"mosout.float\"
    \" $tmpfolder /\ " \"mosout.aiff\" \" $tmpfolder /\ " \"mosout.plot\" )"
echo $init >> $tmpfolder/mos/init/elk/toplevel.scm

# MODALYS_PATH is the path to initialization files
export MODALYS_PATH=$tmpfolder

# Launching Modalys
for files in `pwd`/*.mos; do
bindir/modalys $files
done

# Removing tmpfolder
rm -rf $tmpfolder

# Removing other modalys temp files
for files in /tmp/ld?????;
do
    if ! [ -s $files ] && [ -w $files ];
    then rm -f $files
    fi
done

exit 1
```



11.4 Script Modalys pour XtremWeb - Version Mac OS X

```
#!/bin/sh

export bindir=/Applications/Modalys

# Creating temporary folder for initialization files
tmpfolder=/tmp/$$-modalystmp
mkdir $tmpfolder
mkdir $tmpfolder/mos
cp -r $bindir/mos/ $tmpfolder/mos

# Avoid "normal" initialization of temp files
toplevel_in=$bindir/mos/init/elk/toplevel.scm
toplevel_out=$tmpfolder/mos/init/elk/toplevel.scm
sed -e 's/(mos:initialize /;;;(mos:initialize /' $toplevel_in > $toplevel_out

# Initialize modalys in tmpfolder
init="(mos:initialize-modalys" `\"$tmpfolder/\` \"mosout.float\"
    \"$tmpfolder/\` \"mosout.aiff\" \"$tmpfolder/\` \"mosout.plot\" )"
echo $init >> $tmpfolder/mos/init/elk/toplevel.scm

# MODALYS_PATH is the path to initialization files
export MODALYS_PATH=$tmpfolder

# Launching Modalys
for files in `pwd`/*.mos; do
    $bindir/modalys $files
done

# Removing tmpfolder
rm -rf $tmpfolder

exit 1
```

11.5 Envoi des jobs

```
#!/bin/sh

client_path=/data/xtremweb      # le repertoire ou se trouve XWClient.jar
client_conf_path=/data/xtremweb # le repertoire ou se trouve xwclient.conf
SLEEP=5

function check_client()
{
uname='uname'
case "$uname" in
  "Linux" *) OS="Linux" ;;
  "Darwin" *) OS="Mac-OSX" ;;
  *) echo "unkonwn_OS_(XtremWeb_Client_is_not_supported_for_this_OS)"
     exit 1;;
esac

if [ -e $client_path/XWClient-$OS.jar ]; then break
else echo "Cannot_find_client_file_XWClient-$OS.jar_in_$client_path"
     exit 1
fi

if [ -e $client_conf_path/xwclient.conf ]; then break
else echo "Cannot_find_client_file_xwclient.conf_in_$client_conf_path"
     exit 1
fi

xwclient="java -jar $client_path/XWClient-$OS.jar --xwconfig $client_conf_path/xwclient.conf"
# rq : pour fonctionner correctement, il faut que les droits de l'user soient >= 5
}

function submit_job()
{
# Envoi des jobs
args=$*
JobID=`$xwclient --xwsubmit $args | grep 'successfully submitted' | awk '{print $1}'`
if [ $? -eq 0 ]; then
  echo "Job_$JobID_successfully_submitted_[$args]"
else
  echo "Error_while_submitting_job_$JobID"
return 1
fi

echo $JobID >> /tmp/JobID.txt

# remove job from DB (but not form server disk)
# $xwclient --xwdelete $JobID

return 0
}

function submit_modalys_job()
{
# Envoi simultane des jobs modalys
while [ $i -gt 0 ];do
  submit_job modalys --xwzipfile $1 &

```



```
        i='expr $i - 1'
    shift 1
done
wait

return 0
}

help ()
{
    cat << EOF
HELP

Pour soumettre des jobs sur XtremWeb

USAGE: ./submit_job <application> [Parameters] [<inputFile.txt>]
    Other arguments : --xwzipfile <zipfile> : zipFile to send to the CG platform for this job
                    --xwzip <dir/file> : zip directory/file and send it to the CG platform

    Cas particulier pour Modalys : ./submit_job modalys file1.mos file2.mos ...

OPTIONS: --help message d'aide
        -h message d'aide
        -app liste des applications Ã© installées

EOF

return 0
}

choice=$1

case "$choice" in

    "modalys"|"Modalys")

        shift 1
        i=$#
        if [ -z "$*" ]; then
            echo "Aucun fichier .mos à traiter"
            echo "Veuillez utiliser la syntaxe : ./submit_job .sh modalys file1.mos file2.mos ..."
            exit 0
        else break
        fi

        # Il faut envoyer les fichiers mos en tant que zipfiles
        for files in $*;do
            zip -jq -n .mos $files.zip $files
            modalys_args="$modalys_args " "$files.zip"
        done

        check_client
        submit_modalys_job $modalys_args

        # Suppression des zipfiles
        for files in $*;do
            rm -f $files.zip
        done
    esac
}
```



```
done ;;

"--help"|" -h"|"") help ;;

"--app"|" -app")
  check_client
  echo "Liste des applications installees ::"
  $xwclient --xwapps | awk '{ print $2;}' | sed '/Name/d';

*) check_client
  is_app=`$xwclient --xwapps | grep -w $choice`
  if [ -z "$is_app" ]; then
    echo "Error: unknown command or application"
    echo "Liste des applications installees ::"
    $xwclient --xwapps | awk '{ print $2;}' | sed '/Name/d'
  else
    submit_job $*
  fi
fi
;;
esac

exit 0
```

11.6 Réception des jobs

```
#!/bin/sh

client_path=/data/xtremweb      # le repertoire ou se trouve XWClient.jar
client_conf_path=/data/xtremweb # le repertoire ou se trouve xwclient.conf

function check_client()
{
  uname='uname'
  case "$uname" in
    "Linux"*) OS="Linux" ;;
    "Darwin"*) OS="Mac-OSX" ;;
    *) echo "unkonwn_OS_(XtremWeb_Client_is_not_supported_for_this_OS)"
      exit 1;;
  esac

  if [ -e $client_path/XWClient-$OS.jar ]; then break
  else echo "Cannot_find_client_file_XWClient-$OS.jar_in_$client_path"
    exit 1
  fi

  if [ -e $client_conf_path/xwclient.conf ]; then break
  else echo "Cannot_find_client_file_xwclient.conf_in_$client_conf_path"
    exit 1
  fi

  if [ -e /tmp/JobID.txt ]; then break
  else echo "JobID.txt_introuvable..Aucun_job_effectue?"
    exit 1
  fi

  xwclient="java -jar $client_path/XWClient-$OS.jar --xwconfig $client_conf_path/xwclient.conf"
  # rq : pour fonctionner correctement, il faut que les droits de l'user soient >= 5
}

check_client

SLEEP=1

i=1
JobID='cat -n /tmp/JobID.txt | awk '{ if ($1 == "$i") print $(NF);}' '
while [ ! -z $JobID ];do

  # Recuperation des jobs
  Status="INQUEUE"
  until [ $Status = "COMPLETED" ];do
    sleep $SLEEP
    Status='$xwclient --xwstatus $JobID | awk '{ if ($1 == "$JobID") print $(NF-2); }''
    if [ $Status != "INQUEUE" -a $Status != "RUNNING"
      -a $Status != "COMPLETED" -a $Status = "ERROR" ]; then
      echo "Une_erreur_est_survenue"
      return 1
    fi
  done

  echo "Job_$JobID_completed"
  $xwclient --xwget $JobID
```



```
i='expr $i + 1'  
JobID='cat -n /tmp/JobID.txt | awk '{ if ($1 == "$i") print $(NF);}'`  
done  
rm -f /tmp/JobID.txt  
exit 0
```

11.7 StartupItems pour Mac OS X

Fichier StartupParameters.plist

```
{
  Description      = "XtremWeb_Worker";
  Provides        = ("XWorker");
  Requires        = ("Network");
  Uses            = ();
  OrderPreference = "Last";
}
```

Fichier XWorker

```
#!/bin/sh

. /etc/rc.common

# XtremWeb Worker

StartService ( )
{
  ConsoleMessage "Starting_XtremWeb_Worker"
  nice -n 20 /Applications/XWorker/bin/xtremweb
}

StopService ( )
{
  ConsoleMessage "Stopping_XtremWeb_Worker"
  pid='ps -aux | grep xtremweb.upgrade.Launcher | awk '{ if (NR == 1) print $2}''
  kill -TERM $pid
}

RestartService ( )
{
  StopService
  StartService
}

RunService "$1"
```