

Déployer une réelle alternative à Skype dans nos universités en utilisant des outils libres et standardisés

Emil Ivov*
Jitsi / BlueJimp

Guillaume Schreiner
Université de Strasbourg, Direction Informatique

Philippe Portelli
Université de Strasbourg, Direction des Usages du Numérique

Résumé

Aujourd'hui, l'utilisation de Skype est déconseillée dans nos établissements et laboratoires. Dans la réalité, cette directive du Ministère de l'Enseignement et la Recherche est rarement appliquée et nombreux sont les utilisateurs de ce produit pour une raison simple : le manque d'alternatives.

Cependant, une réelle solution basée sur des outils libres et standardisés est possible et relativement simple à réaliser dans la plupart des universités françaises. Sur cette idée, l'Université de Strasbourg (UdS) a lancé un projet ayant pour but de fournir un service de messagerie instantanée et de visioconférence à destination de l'ensemble des étudiants, enseignants, chercheurs et administratifs.

Cet article décrit en détail les différentes améliorations apportées au logiciel de messagerie instantanée Jitsi et son intégration au système d'information de l'UdS. Nous aborderons ensuite l'infrastructure composant le service qui utilise les serveurs Openfire, TurnServer, JingleNodes, FreeSWITCH ainsi que leur intégration respectant les contraintes de haute disponibilité. Enfin, nous présenterons les différents services que ce projet propose à nos utilisateurs.

Mots clefs

VoIP, visio-conférence, messagerie instantanée, SIP, IPv6

1 Introduction

Aujourd'hui, l'utilisation de Skype est interdite ou fortement déconseillée dans la plupart des établissements publics en France. Les raisons sont nombreuses et variées, allant des risques de sécurité et d'espionnage industriel à la surconsommation de bande passante. Néanmoins, dans la réalité, ces conseils sont rarement appliqués et nous continuons tous à nous servir de Skype pour une raison principale : le manque d'alternatives.

En effet, peu d'applications aujourd'hui regroupent les mêmes avantages :

- gratuit à l'utilisation
- appels audio et vidéo de bonne qualité
- conférences
- support des principaux systèmes d'exploitation (Windows, Mac OS X et Linux)
- sécurité : des appels chiffrés

* Pour toute question ou commentaire sur cet article, veuillez contacter Emil Ivov: emcho@jitsi.org

- [particulièrement important] une large communauté d'utilisateurs, facilitant son adoption
- facilité d'utilisation

Ces caractéristiques ont rendu Skype indispensable pour la plupart des collaborations qui ont lieu dans les universités et les laboratoires ainsi que pour des conversations personnelles.

Cependant, une vraie alternative basée sur des outils libres est plus réaliste que certains ne l'imaginent. Dans le cadre du schéma directeur numérique de l'établissement, l'Université de Strasbourg (UdS) a démarré en 2011 un projet ambitieux dont le but est de fournir une alternative concrète pour l'ensemble des étudiants, enseignants, chercheurs et administratifs. Du côté de l'université, l'effort est à l'initiative de la Direction des Usages du Numérique avec le soutien de la Direction Informatique. Le projet se fait en étroite collaboration avec la société BlueJimp qui est le principal contributeur au projet open source Jitsi (précédemment nommé SIP Communicator).

Dans cet article, nous allons nous attacher à décrire les développements et les déploiements effectués dans le cadre du projet. Il fait suite à une première présentation [2], lors de l'édition 2007 des JRES, du projet SIP Communicator (parent de Jitsi) qui représente une des bases de notre alternative à Skype. Ce nouvel article représente donc une suite logique ainsi que l'inscription du projet dans un contexte réel d'utilisation.

2 Rappels protocolaires

Pour des contraintes d'espace, certains des mécanismes utilisés dans le projet d'alternative Skype ont été présentés de manière assez sommaire, une version complète de l'article [1] étant également disponible pour le téléchargement. Voici quelques rappels indispensables à propos des différents protocoles mis en œuvre.

2.1 XMPP

Dans le contexte du projet, XMPP [3] (Extensible Messaging and Presence Protocol, à l'origine appelé Jabber) est la pierre angulaire protocolaire. Initialement, le protocole se focalise sur la présence et la messagerie instantanée. Contrairement aux solutions populaires de l'époque, telles que *ICQ* ou *Yahoo! Messenger*, Jabber a depuis ses premières versions été conçu pour fonctionner dans un contexte multi-domaines. Comme pour le mail, chaque utilisateur est associé à un domaine administratif sans que cela n'empêche les communications avec le reste du monde. Dans le jargon du milieu, cette connexion inter-domaines est souvent appelée « fédération » et, comme avec la messagerie électronique, elle passe par l'utilisation d'identifiants de type « URI » comme *xmpp:alice@example.com* et *xmpp:bob@exemple.fr*.

Contrairement à SMTP, la communauté de XMPP a proposé un modèle de sécurité assez robuste dès son déploiement. En effet, le protocole impose à chaque serveur l'utilisation de TLS et de certificats SSL, ce qui réduit largement le risque d'utilisations abusives telles que le spam.

Pendant les années 2000, les adoptions industrielles ne cessent de se multiplier. L'exemple le plus connu est celui de Google avec le service GoogleTalk. Peu après le lancement initial du service en 2005, Google y ajoute le support des appels audio/vidéo. Les extensions permettant cette fonctionnalité, appelées Jingle, sont ensuite rendues publiques par Google qui poursuit son développement en collaboration avec la XMPP Standards Foundation (XSF). En 2009, une première version standardisée de Jingle [4] est publiée par la XSF. XMPP dispose désormais d'une palette complète d'outils pour les communications temps réel.

2.2 La traversée des NATs

Les protocoles SIP et XMPP utilisent un modèle trapézoïdal de communication entre deux utilisateurs. Chaque utilisateur est connecté au serveur responsable de son domaine. Les messages de signalisation, comme l'initiation, la modification ou la terminaison des appels, traversent ces serveurs. L'essentiel du trafic par contre, celui transportant les données audio ou vidéo, est sensé circuler directement entre les parties communicantes. Malheureusement, après sa conception, ce modèle s'est très vite avéré incompatible avec la réalité en raison de l'utilisation grandissante de la translation d'adresses (NAT).

* Pour toute question ou commentaire sur cet article, veuillez contacter Emil Ivov: emcho@jitsi.org

Durant les dix dernières années, l'industrie et la recherche ont proposé de multiples mécanismes visant à contourner ce problème. Les plus importants d'entre eux sont présentées en détail dans la version complète [1] de cet article. Dans le cadre de ce projet nous nous basons principalement sur le protocole ICE [5] (Interactive Connectivity Establishment). Il permet l'établissement de connexions de bout en bout et de manière interactive tout en minimisant le besoin de relayer des communications.

2.3 Sécurité et chiffrement des appels avec ZRTP

En 2006, Phil Zimmerman propose un nouveau mécanisme de sécurisation des appels VoIP appelé ZRTP [7]. Le protocole repose sur l'utilisation de paires de clés privées et publiques et la création d'un secret partagé à travers un échange Diffie-Hellman. ZRTP se démarque des autres alternatives grâce à l'authentification et la prévention des attaques de type « *Man In The Middle* » (MITM). Pour assurer les participants que leurs données sont envoyées et reçues uniquement par eux-mêmes, ZRTP affiche un *checksum* dérivé du secret partagé aussi connu sous le nom SAS (*Short Authentication String*). Ce *checksum* apparaît le plus souvent en tant que quatre lettres dans l'interface de l'agent. Les utilisateurs sont tout simplement sensés vérifier que chaque participant voit les mêmes quatre lettres, en les prononçant pendant leur premier appel. Pour continuer son écoute, un attaquant MITM, devrait donc à ce moment arriver à remplacer les paroles des utilisateurs par d'autres, tout en imitant leur voix. Cette vérification est nécessaire uniquement lors de la première conversation avec un utilisateur donné et ne sera plus effectuée lors des futures conversations. Bien évidemment, l'utilisation de ZRTP implique une certaine prise en main de la part des utilisateurs qui pourrait être rébarbative. Or, vu le haut niveau de sécurité garanti par ZRTP et l'absence totale de coût de déploiement, on peut affirmer que la solution est parmi les plus raisonnables existantes aujourd'hui.

3 Le projet libre Jitsi

Le projet libre Jitsi représente une des principales composantes de la solution décrite dans ce document. Il s'agit d'une application cliente permettant les appels vidéo, le partage de bureau, la messagerie instantanée, le transfert de fichiers et plusieurs autres fonctionnalités que nous allons évoquer en détail ci-dessous.

Certains des lecteurs connaissent peut-être l'application sous son ancien nom: SIP Communicator. Une des premières versions du logiciel a d'ailleurs déjà été présentée lors des JRES 2007 [2]. Dans cette section, nous allons surtout nous concentrer sur les nouvelles fonctionnalités que nous trouvons essentielles pour le déploiement à l'Université de Strasbourg.

3.1 Historique

Jitsi, appelé SIP Communicator à ses débuts, démarre en 2003 comme projet étudiant à l'Université de Strasbourg. Le logiciel reste dans un état expérimental pendant quelques années. En 2007, Emil Ivov, fondateur de Jitsi (et l'un des auteurs de cet article), lance avec quelques collaborateurs un projet d'entreprise innovante ayant pour idée de proposer de la maintenance et des intégrations professionnelles du logiciel. La nouvelle start-up BlueJimp collabore avec plusieurs partenaires et donateurs, tels que la NLnet Foundation (Pays-Bas), l'Université de Strasbourg, le programme Google Summer of Code et d'autres. De fil en aiguille, Jitsi devient ce qu'il est aujourd'hui : l'un des logiciels communicants libres les plus aboutis et des plus riches en fonctionnalités.

3.2 Fonctionnalités

Avant le démarrage du projet à l'UdS, Jitsi comptait déjà un nombre important de fonctionnalités. Il y en a cependant beaucoup qui ont été ajoutées explicitement dans le contexte du projet de déploiement et dont le financement a été assuré par l'UdS. Aujourd'hui, elles sont toutes disponibles librement pour ceux qui souhaitent tenter l'expérience. Voici maintenant les plus innovantes.

* Pour toute question ou commentaire sur cet article, veuillez contacter Emil Ivov: emcho@jitsi.org

3.2.1 Appels audio/vidéo de haute qualité

Les appels vidéo sont certainement l'une des fonctionnalités phares de Jitsi. Les appels sont par défaut établis avec une résolution VGA (640x480 pixels) et il est possible pour l'utilisateur de les modifier ce qui peut s'avérer pratique dans le cas d'un débit réseau insuffisant. Le codec utilisé par défaut est H.264 et Jitsi supporte également H.263.

En ce qui concerne l'audio, les équipes de BlueJimp ont ajouté dans le contexte du projet UdS le support pour plusieurs codecs de haute qualité tels que Silk, G.722, capable de fonctionner sur une fréquence de 16 kHz, 24 kHz et 32 kHz. Il est important de noter que le codec Silk est celui qui est également utilisé par Skype. On obtient donc exactement la même qualité audio en utilisant Jitsi que lors des fameux appels HD de Skype.

3.2.2 Support de SIP, XMPP et d'autres protocoles

Jitsi propose un support complet des fonctionnalités de présence et de messagerie instantanée pour un grand nombre de protocoles et de réseaux comme: Windows Live (MSN), Facebook, Google Talk, Yahoo! Messenger, ICQ, AIM et bien entendu SIP et XMPP.

Les fonctionnalités de téléphonie et d'appels vidéo sont actuellement proposées pour SIP mais aussi (et c'est assez rare) pour XMPP et GoogleTalk sur les systèmes d'exploitation Windows, Linux et Mac OS X. Le support de la variante Google de Jingle a été développé dans le contexte du projet actuel dans le but de l'ouvrir à une large communauté d'utilisateurs. Les utilisateurs qui ne sont pas enregistrés sur le domaine de l'UdS sont joignables avec leur compte Gmail ou XMPP.

3.2.3 Traversée des NATs

La communauté de Jitsi ainsi que les équipes de BlueJimp ont fourni un grand effort pour l'implémentation des protocoles STUN, TURN, et JingleNodes dans une implémentation complète de ICE dans le cadre du projet ice4j.org [8]. Cette pile intègre également l'utilisation d'autres mécanismes de traversée des NATs comme UPnP, et les relais TCP. Jitsi se sert d'ice4j.org pour assurer la traversée des NATs avec XMPP. En ce qui concerne SIP, Jitsi dépend des serveurs SIP pour assurer le relaying. L'intégration de ICE pour SIP est prévue pour la fin de l'année.

3.2.4 Appels de conférence

Le plus souvent, pour participer à des appels de conférences il est nécessaire pour les utilisateurs de joindre un numéro spécial (un pont de conférence). Une fois la connexion établie, les différents participants sont mis en relation de manière transparente mais, même s'ils arrivent tous à s'entendre, le logiciel ou l'équipement qui a effectué l'appel pour l'utilisateur le perçoit comme un simple appel point à point ce qui rend impossible la présentation visuelle des autres participants ainsi que leur activité dans l'appel.

Bien que Jitsi puisse participer à ce genre d'appels, il est également capable de gérer les informations envoyées par le serveur pour afficher aux utilisateurs une représentation plus complète de l'appel (voir l'image ci jointe Figure 2).

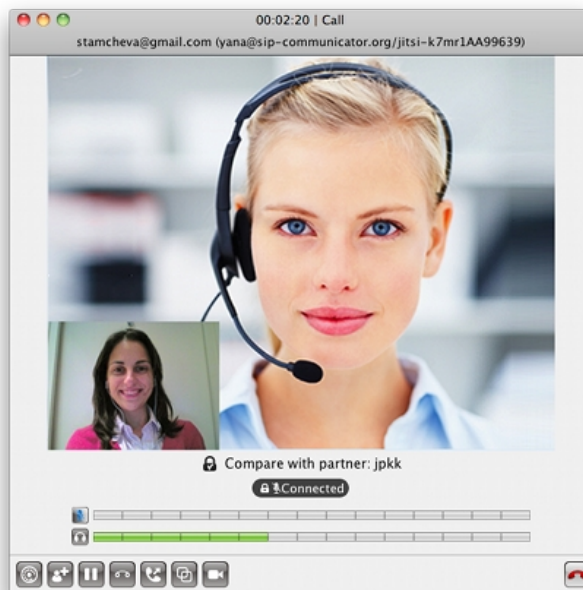


Figure 1: Appel vidéo avec Jitsi

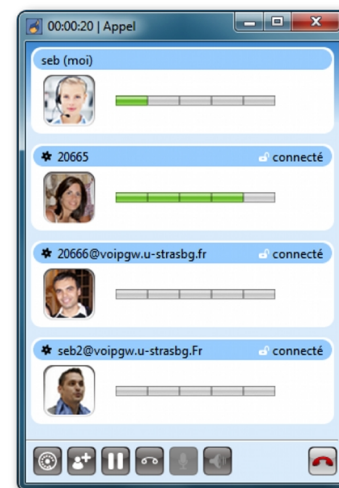


Figure 2: Appel de conférence

* Pour toute question ou commentaire sur cet article, veuillez contacter Emil Ivov: emcho@jitsi.org

Plus important encore, Jitsi est capable d'héberger lui-même des appels de conférences de manière entièrement indépendante du serveur et sans limite pour le nombre maximal de participants.

3.2.5 Sécurité et confidentialité

Pour ses appels audio aussi bien que vidéo, Jitsi supporte la sécurisation avec SRTP et les échanges des clés avec SDES pour SIP et ZRTP pour SIP et XMPP. Cette combinaison est assez rare en ce qui concerne ZRTP, Jitsi est le seul logiciel, à notre connaissance, qui propose son support pour les conversations effectuées avec XMPP.

3.2.6 Divers

En plus des points énumérés ci-dessus, d'autres se sont avérés assez importants dans le contexte du projet UdS. Jitsi a la particularité d'être **multi-plateformes**. La communauté supporte actuellement des versions pour Windows, Linux et MacOS X. De par son parc informatique très hétérogène, le support des différents systèmes d'exploitation était indispensable pour l'université. Enfin, le logiciel propose également l'affichage à distance ou le partage du bureau, les consultations d'un annuaire LDAP et la configuration à distance (*online provisioning*).

4 Infrastructure du service

4.1 Service XMPP avec Openfire

Comme nous l'avons déjà mentionné, le protocole XMPP avec sa sécurité, son extensibilité et sa richesse en fonctionnalités, a été choisi comme la base du service. Il existe plusieurs serveurs XMPP libres et de très bonne qualité comme par exemple *Openfire* [9], *ejabberd* [10] et *Prosody* [11].

Dans le cadre du projet à l'UdS, nous nous sommes arrêtés sur Openfire entre autres pour sa flexibilité et sa communauté.

L'authentification des utilisateurs se fait entre le serveur XMPP Openfire et l'annuaire LDAP. À chaque connexion d'un client, Openfire interroge l'annuaire et vérifie les droits d'accès. De cette manière, il n'est pas nécessaire de dupliquer la base utilisateurs. Si un nouvel arrivant est ajouté au système d'information, il est automatiquement ajouté au serveur LDAP et peut directement se connecter au service sans aucune manipulation.

Finalement, pour faciliter la mise en place du service, le client Jitsi a été modifié pour proposer une première configuration simplifiée. Pour qu'un utilisateur commence à utiliser le nouveau service, il lui suffira de télécharger le logiciel et de rentrer le nom et le mot de passe de son compte habituel. Ceci le connectera au service XMPP et configurera les propriétés nécessaires pour l'utilisation de l'annuaire LDAP et du serveur STUN de l'université.

4.2 Traversée des NATs et des pare-feux

Le service de communication décrit ici a pour but de permettre aux étudiants et personnels de l'Université de Strasbourg de communiquer entre eux et avec leurs contacts à n'importe quel moment et n'importe quel endroit. Autrement dit, même s'il est vrai que l'Université de Strasbourg dispose d'un nombre confortable d'adresses IPv4 publiques, les utilisateurs du service XMPP ou leurs interlocuteurs peuvent se retrouver dans des topologies diverses. La traversée des NATs doit donc être assurée de manière fiable.

De plus, lors des premiers tests d'appels, il s'est avéré que les communications directes sont interdites entre les utilisateurs connectés au réseau sans fil de l'UdS. À l'origine, la mesure avait été mise en place dans le but d'éviter les dysfonctionnements ou les attaques basées sur DHCP ou IPv6 RA. L'absence de moyens plus fins pour la configuration de l'infrastructure sans fil avait donc obligé les équipes de la direction informatique à interdire toute communication entre des nœuds voisins. Évidemment, cette interdiction ne pose pas de problème pour les échanges web ou mail, mais elle produit un effet de bord gênant dans le cas des appels IP. Elle rend donc indispensable le déploiement de serveurs relais.

* Pour toute question ou commentaire sur cet article, veuillez contacter Emil Ivov: emcho@jitsi.org

La solution prévue initialement pour ces relais était l'utilisation du protocole TURN et du serveur libre TurnServer.org, car c'est la solution de relaying recommandée par l'IETF pour l'utilisation avec ICE. Malheureusement, il est vite devenu clair que cette solution ne serait pas applicable dans l'environnement de l'UdS pour des raisons détaillées dans la version complète de cet article [1]. Nous avons donc préféré utiliser les relais JingleNodes, disponibles en tant que plug-in dans le serveur Openfire.

4.3 Appels vers des fixes

Un des objectifs du projet d'alternative à Skype est de pouvoir initier des communications audio depuis le logiciel client Jitsi vers tous les postes de téléphone classiques gérés par l'université, afin de pouvoir joindre des utilisateurs n'ayant pas d'ordinateur ou n'ayant pas le client Jitsi installé ou démarré.

L'infrastructure téléphonique de l'Université de Strasbourg est composée d'un parc téléphonique hétérogène où cohabitent plusieurs technologies de générations différentes. La majorité du parc téléphonique est composée de lignes analogiques et numériques reliées à des PABX Alcatel. Une des premières solutions envisagées consistait à raccorder un serveur SIP au réseau téléphonique classique en utilisant une carte Zaptel [12] et de l'utiliser en tant que passerelle vers les utilisateurs de Jitsi. Cette solution a été écartée pour 3 raisons. Tout d'abord, la politique universitaire de virtualisation des serveurs se base sur un modèle de serveur physique standard dans le but de déplacer les machines virtuelles de manière transparente. Une carte matérielle supplémentaire compliquait fortement cette souplesse d'administration. Ensuite, une solution analogique nécessitait le raccordement du serveur physique au PABX via des câbles téléphoniques dédiés. Dans un futur proche, un déplacement géographique des machines vers un site dépourvu de PABX rendrait donc cette solution obsolète. Enfin, cette solution n'est pas pérenne car l'université souhaite à moyen terme migrer entièrement vers de la téléphonie sur IP. Bien entendu toutes ces raisons étant spécifiques à l'UdS, cette solution reste techniquement viable et peut être envisagée dans d'autres circonstances.

Revenons au cas de l'Université de Strasbourg. Il s'est avéré que parmi les PABX téléphoniques, certains (Alcatel OmniExpress OXE plus précisément) disposent d'interfaces SIP. Une connexion physique n'est donc plus nécessaire et peut se faire par le réseau IP. Il est donc possible pour un serveur SIP d'utiliser ces PABX comme point d'entrée vers le réseau téléphonique et d'y acheminer des appels provenant du réseau IP.

Dernier point à résoudre : comment établir la connexion entre les utilisateurs Jitsi et ces serveurs SIP ? Une idée relativement simple serait de connecter les clients Jitsi, supportant eux même le protocole SIP, directement au serveur. Malheureusement, ceci présente plusieurs difficultés. Tout d'abord, une telle connexion signifierait l'existence de plusieurs comptes et la maintenance de multiples connexions simultanées au niveau du client. Elle serait relativement complexe à mettre en place et éloignerait la solution des standards. La présentation à l'utilisateur, dans l'interface graphique, de comptes distincts SIP et XMPP pourrait être incomprise et devenir perturbante pour l'utilisateur qui a, lui, fait le choix d'utiliser un seul service.

Plus important encore, étant donné que SIP utilise le même modèle d'authentification que le protocole TURN, l'authentification au niveau du serveur aurait présenté exactement les mêmes difficultés que l'utilisation d'un serveur TURN. Les mots de passe utilisateurs stockés dans le LDAP sont au format SHA1 et ne sont donc pas exploitables par les serveurs SIP ou TURN.

Nous avons donc cherché une autre solution qui permettrait aux clients Jitsi de garder une seule connexion de signalisation : celle avec le serveur XMPP et qui ne demanderait pas d'authentification au niveau du serveur SIP. Là encore le protocole XMPP et son modèle sécuritaire se sont avérés très utiles.

Le serveur SIP FreeSWITCH [13] bénéficie d'une fonctionnalité qui lui permet de s'enregistrer en tant que composante dans un serveur XMPP et de gérer les sessions Jingle. Autrement dit, si le serveur XMPP dessert le domaine *unistra.fr* et que le serveur FreeSWITCH s'enregistre comme composante sous le nom *freeswitch*, toute demande d'appel Jingle vers des adresses comme exemple@freeswitch.unistra.fr sera transmise au serveur FreeSWITCH.

* Pour toute question ou commentaire sur cet article, veuillez contacter Emil Ivov: emcho@jitsi.org

Il est important de noter que même si la signalisation traverse toujours le serveur XMPP, le trafic RTP est échangé directement entre les clients Jitsi et le serveur FreeSWITCH.

La Figure 3 résume les différentes interactions entre le client Jitsi et les briques logicielles du service.

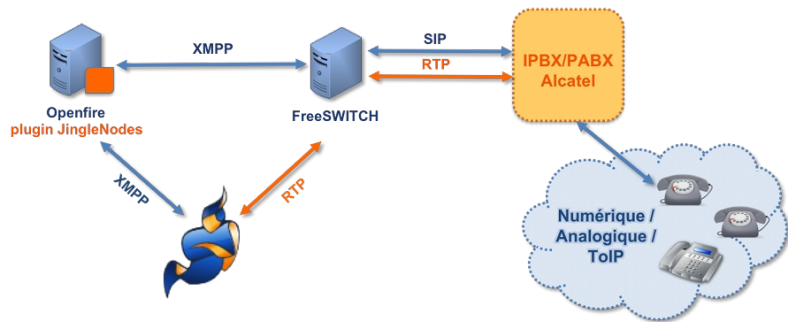


Figure 3: Appels vers des fixes

4.4 Virtualisation et redondance

Dans la mouvance actuelle, chaque serveur de la Direction Informatique à l'UdS est virtualisé si les applications le permettent. Les avantages de la virtualisation en terme d'administration sont les suivants :

- mutualisation de serveurs physiques
- déplacement de machines virtuelles à chaud
- reprise sur panne matérielle plus rapide
- adaptation des ressources à la demande (CPU RAM)

Les serveurs de virtualisation à l'UdS sont des machines de 8 à 24 cœurs avec des CPU de type Intel Xeon X5550 à 2.67 GHz. La mémoire vive disponible est comprise entre 24 et 64 Go. Les images des disques des machines virtuelles sont stockées sur des disques durs locaux montés en RAID 6 avec un contrôleur matériel.

Le système standard de virtualisation est KVM déployé sur le système GNU/Linux Ubuntu LTS.

Dans le cadre du projet Jitsi, la solution logicielle retenue est suffisamment souple pour se prêter à la virtualisation. Deux machines virtuelles assurent une redondance active/passive à l'aide du logiciel Pacemaker [14]. Le serveur maître possède l'adresse IP virtuelle du service Jitsi et lance automatiquement l'ensemble des démons nécessaires (Openfire, TurnServer, FreeSWITCH, MySQL). En cas de défaillance du serveur maître (panne, maintenance), le serveur esclave se réveille, récupère l'adresse IP virtuelle et relance les services. La base de données MySQL est un point particulier car elle tourne en local sur le serveur maître. Cette base est couplée à un volume réseau DRBD [15] répliqué en temps réel entre les deux serveurs. Ce fonctionnement peut être assimilé à un montage RAID 1 en réseau.

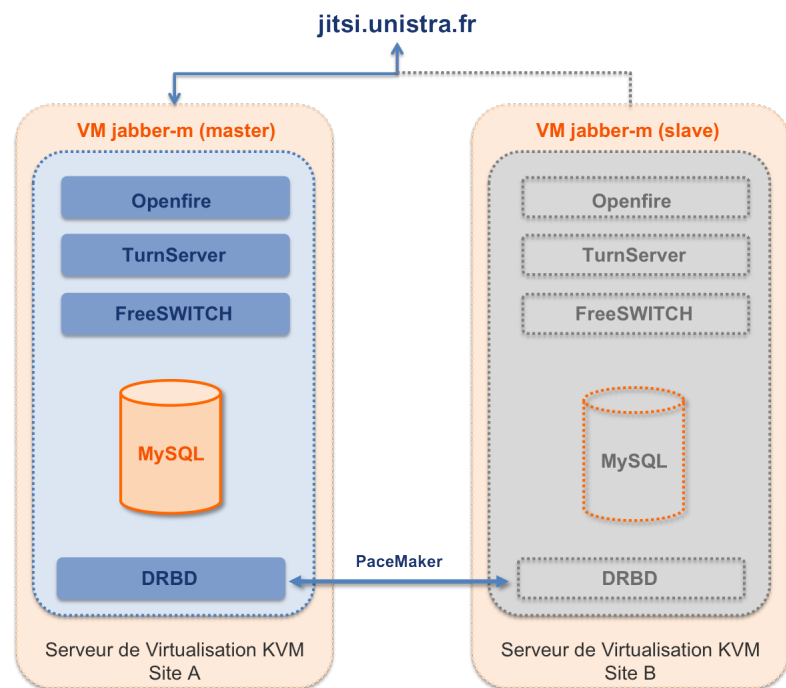


Figure 4: Virtualisation, redondance et montée en charge

Ainsi, en relançant MySQL sur le serveur esclave, les dernières données écrites sur le serveur maître sont disponibles.

La Figure 4 résume l'architecture et la redondance du service déployé.

* Pour toute question ou commentaire sur cet article, veuillez contacter Emil Ivov: emcho@jitsi.org

5 Conclusion et perspectives d'évolution

Comme évoqué dans l'introduction de cet article, en commençant le projet de communication temps réel à l'UdS, nous avons identifié sept points qui donnent à Skype sa popularité actuelle et qui devront être présents dans toute solution cherchant à le remplacer. Aujourd'hui, une fois le projet terminé, il est intéressant de revenir sur cette liste et d'évaluer l'alternative déployée suivant les différents critères.

Gratuit à l'utilisation – il est évident qu'avec les solutions utilisées (Jitsi, Openfire, FreeSWITCH, TurnServer et JingleNodes), nous avons un service qui va plus loin que la simple gratuité. Toutes les composantes du projet sont distribuées sous une licence libre et peuvent être reprises, déployées ou modifiées par tout le monde, sans aucune contrainte commerciale.

Appels audio et vidéo de bonne qualité – avec le support des codecs audio de haute qualité tels que G.722, Speex et même SILK utilisé par Skype, ainsi que les appels vidéo en résolution 640x480, les appels avec Jitsi et le nouveau service sont tout à fait comparables et souvent meilleurs que ceux effectués avec Skype.

Conférences – avec Jitsi, qui peut fonctionner en tant que mixeur audio et agent de focus, il est aujourd'hui possible pour les utilisateurs du service d'organiser ou de participer à des appels de conférence avec, encore une fois, une très bonne qualité audio.

Support des principaux systèmes d'exploitation (Windows, Mac OS X et Linux) – vérifié avec Jitsi, qui sera également bientôt disponible en version Android. De plus, l'utilisation exclusive de solutions standards permet aux utilisateurs d'utiliser tout client supportant les mêmes protocoles.

Sécurité : appels chiffrés – l'utilisation de SSL dans XMPP et le support de ZRTP dans Jitsi garantissent une sécurité nettement meilleure que celle de Skype, où l'intimité des utilisateurs est soumise à la bonne volonté du fournisseur du service.

Une large communauté d'utilisateurs, facilitant son adoption – avec le support des variantes Google de Jingle, il est possible aujourd'hui pour les utilisateurs UdS d'effectuer des appels vers tous les abonnés XMPP ou Gmail/GTalk qui, dans leur ensemble, représentent une communauté comparable à celle des utilisateurs Skype.

Facilité d'utilisation – cette caractéristique étant relativement subjective, nous invitons les lecteurs à l'évaluer par eux même en se rendant sur <http://jitsi.org>

Bien évidemment, nous ne comptons pas en rester là. Dans le mois à venir, le service devrait être étendu pour permettre les appels vidéo à plusieurs participants (fonctionnalité payante avec Skype).

Des solutions de facturation plus avancées sont également à l'étude et devraient permettre au personnel de l'UdS d'effectuer des appels vers tous les postes téléphoniques ou mobiles et non plus seulement vers ceux appartenant à l'université.

Nous espérons que cet exemple de déploiement d'alternative à Skype encouragera d'autres universités à franchir le pas, créant ainsi une fédération de réseaux interopérables pour les utilisateurs.

6 Remerciements

Les auteurs aimeraient remercier Yana Stamcheva (Jitsi/BlueJimp) pour avoir fourni les diagrammes ainsi que Véronique Dupont (Alcatel-Lucent), Guillaume Lacroix (ippi.fr) et Eric Laemmer (Université de Strasbourg) pour leur relecture et commentaires.

7 Bibliographie

- [1] Emil Ivov, Guillaume Schreiner et Philippe Portelli, Déployer une réelle alternative à Skype dans nos universités en utilisant des outils libres et standardisés, <http://bluejimp.com/whitepapers/jres2011-complete-fr.pdf>
- [2] Emil Ivov et Jean-Marc Muller, SIP Communicator – Un outil open source de communication sur IP adapté à nos laboratoires et à nos universités. *Journées Réseaux*, Novembre 2007.

* Pour toute question ou commentaire sur cet article, veuillez contacter Emil Ivov: emcho@jitsi.org

- [3] Peter Saint-Andre, RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core. *Internet Engineering Task Force*, March 2011.
- [4] Scott Ludwig, Joe Beda, Peter Saint-Andre, Robert McQueen, Sean Egan, Joe Hildebrand, XEP-0166: Jingle. XMPP Standards Foundation, Décembre 2009.
- [5] Jonathan Rosenberg, Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols. *Internet Engineering Task Force*, Avril 2010.
- [6] F. Andreasen, M. Baugher, D. Wing, RFC 4568: Session Description Protocol (SDP) Security Descriptions for Media Streams. *Internet Engineering Task Force*, July 2006.
- [7] P. Zimmermann, A. Johnston, J. Callas, RFC 6189: ZRTP: Media Path Key Agreement for Unicast Secure RTP. *Internet Engineering Task Force*, April 2011.
- [8] Communauté Jitsi, Implémentation Java du protocole ICE, <http://ice4j.org>
- [9] Openfire, a real time collaboration server, <http://www.igniterealtime.org/projects/openfire/>
- [10] A Jabber/XMPP instant messaging server. <http://www.ejabberd.im/>
- [11] Communications server for Jabber/XMPP, <http://prosody.im/>
- [12] ZapTel Calling Cards, <http://www.zaptel.com/>
- [13] FreeSWITCH, open source cross-platform telephony platform, <http://freeswitch.org/>
- [14] Pacemaker, Open Source, High Availability resource manager, <http://www.clusterlabs.org/>
- [15] DRBD, block devices designed as a building block, <http://www.drbd.org>

* Pour toute question ou commentaire sur cet article, veuillez contacter Emil Ivov: emcho@jitsi.org