EURANOVA R&D

Euranova est une société Belge constituée depuis le 1er Septembre 2008. Sa vision est simple : « Être un incubateur technologique focalisé sur l'utilisation pragmatique des connaissances ».

Les activités de recherche sont associées à des voies technologiques et à des opportunités concrètes sur le court, moyen et long terme.

Euranova découple la gestion de carrière de la relation client en s'appuyant notamment sur une perception entrepreneuriale de la carrière.

OBJET DU DOCUMENT

Nous proposons ici des mémoires dans notre département Recherche & Développement. L'étudiant travaillera en collaboration avec les ingénieurs de recherche et sera amené à partager ses travaux à travers l'outil de gestion de la connaissance utilisé en interne par Euranova.

CONTENU

DESCRIPTION DES MÉMOIRES

ETUDE ET IMPLÉMENTATION D'UNE ARCHITECTURE EDA POUR LE SUIVI EN TEMPS RÉEL DE PATIENTS D'UNITÉS DE SOINS INTENSIFS

Contexte : L'aide à la décision informatisée est une science en pleine croissance que l'on retrouve dans différents secteurs mais pas encore dans tous. En effet, dans les unités de soins intensifs, les capteurs qui mesurent en temps réel l'état des patients envoient une grande quantité de données aux médecins, comme la pression sanguine, le taux de créatine, le taux d'anticorps, etc. Aujourd'hui il n'existe pas d'infrastructures automatisées permettant de collecter l'ensemble de ces événements afin d'en inférer une situation devant requérir l'intervention des équipes médicales.

Des équipes de recherches se sont attaquées au problème et ont proposé des solutions basées sur le serveur d'application temps réel JAIN SLEE et un ESB [7]. Ce prototype permet d'implémenter des logiques de surveillance pouvant détecter des situations problématiques pour un patient. Cependant cette architecture n'est pas la solution idéale, principalement pour trois raisons :

- Le *container* JSLEE doit maintenir une session *state full* par patient, ce qui pourrait être un frein à la montée en charge pour supporter un grand nombre de patients,
- Nécessite d'implémenter la logique de détection en JAVA, nous ne pouvons donc pas utiliser de *Domain Specific Languages* (DSL) accessibles au monde médicale pour générer les règles,
- L'architecture n'est pas évolutive ex : Comment coupler des données contextuelles temps réel, comme la présence d'un spécialiste, la présence d'une infermière, ou d'autres données, aux logiques de détection de problèmes ?

Nous proposons d'implémenter une solution de surveillance des patients basée sur les architectures orientées événements (EDA) et l'intégration d'un moteur d'inférence temps réel de type CEP (*Complex Event Processor*).

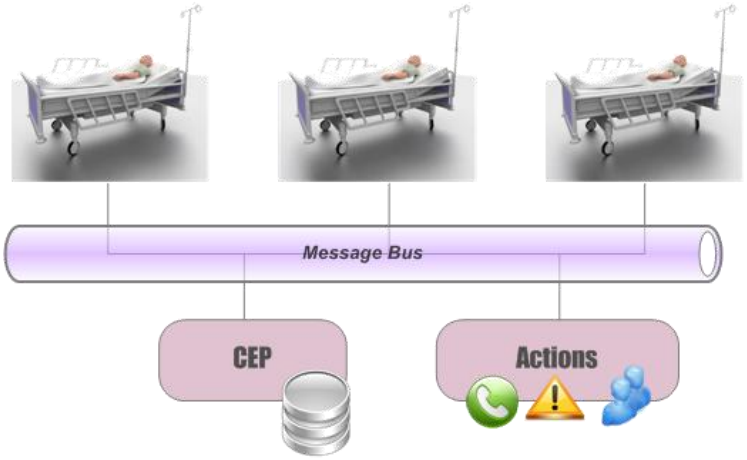

Figure 1 The EDA used for the real time patient monitoring.

Contribution : L'objectif du mémoire est (1) d'étudier, via une série d'interviews de médecins d'unités des soins intensifs, les différents paramètres de surveillance des patients ainsi que quelques règles simples de détection de problèmes, (2) d'étudier les architectures orientées événements et un moteur CEP en particulier, (3) de proposer une architecture EDA et enfin (4) d'implémenter un prototype EDA supportant 50 instances de simulateurs de patients ayant chacun des scénarios de tests XML.

Organisation: ce mémoire est organisé par l'ULB en collaboration avec Euranova R&D. L'étudiant sera accompagné par l'équipe d'Euranova R&D.

ETUDE ET IMPLÉMENTATION D'UNE CACHE L2 POUR MOBICENTS JSLEE

Contexte : Aujourd'hui la plupart des serveurs d'application JEE utilise des niveaux de cache L1-L2 pour permettre un déploiement en *cluster* plus aisé et ainsi supporter une montée en charge importante. Différents frameworks peuvent être utilisés à cette fin, citons par exemple JBoss cache, INFINISPAN et Terracotta.

Mobicents JSLEE est quant à lui un serveur d'application JAVA open source orienté message et temps réel implémenté par Red Hat dans le cadre de la *Communication Media Platform*. Le but de cette plateforme est de pouvoir créer, déployer, et gérer des services et applications qui intègrent la voix, la vidéo et les données sur les réseaux IP et de communications.

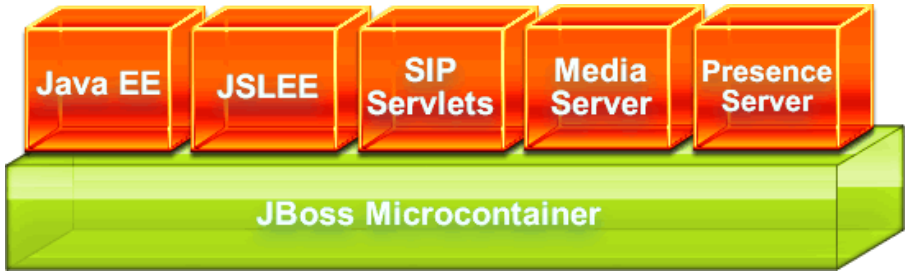

Figure 2 The Mobicents communication platform [1]

La dernière version de Mobicents propose une clusterisation du serveur mais pas encore un niveau de cache pour les accès en base de données, ce qui crée un important goulot d'étranglement lors de la montée en charge.

Contribution : Le but de ce mémoire est d'étudier (1) le container JSLEE de Red Hat et plus particulièrement l'accès aux bases de données à travers les profiles, (2) les systèmes de caches distribués, (3) de proposer une architecture de cache L2 et enfin (4) d'implémenter un prototype.

Organisation: ce mémoire est organisé par l'ULB en collaboration avec Euranova R&D. L'étudiant sera accompagné par l'équipe d'Euranova R&D mais également par les contacts auprès de l'équipe d'ingénierie de Mobicents chez Red Hat.

ETUDE ET COMPARAISON DES BASES DE DONNEES ELASTIQUES SUR CLOUD : UN MYTHE ?

Contexte : Aujourd'hui, le *cloud computing* est certainement un des sujets les plus à la mode. En effet, il promet une évolution élastique de nos serveurs d'applications juste par quelques simples clics sur l'interface de management en y ajoutant dynamiquement de nouvelles machines virtuelles. Mais le *cloud* est-il vraiment élastique pour tout type d'application, y compris celles qui possèdent une importante quantité de données ?

Amazon propose différentes formules de stockage de données sur EC2 : les *Elastic Blocs Storage* (EBS) et les fichiers S3. Ces solutions n'offrent pas une grande flexibilité : EBS requiert des *backups* dont l'interruption de service n'est pas contrôlée (verrous pendant les fenêtres de *backup*), de plus les EBS ne peuvent être montés que pour un seul nœud, ce qui expose un *single point of faillure*. Amazon *Simple Storage Service* (Amazon S3) offre le stockage de données via des *Webservices*, ce qui implique un temps de latence important surtout lorsque les écritures sont fréquentes. Depuis quelques mois d'autres vendeurs arrivent sur le *cloud* d'Amazon, mais sont-ils plus élastiques et sur quelles architectures reposent-ils ?

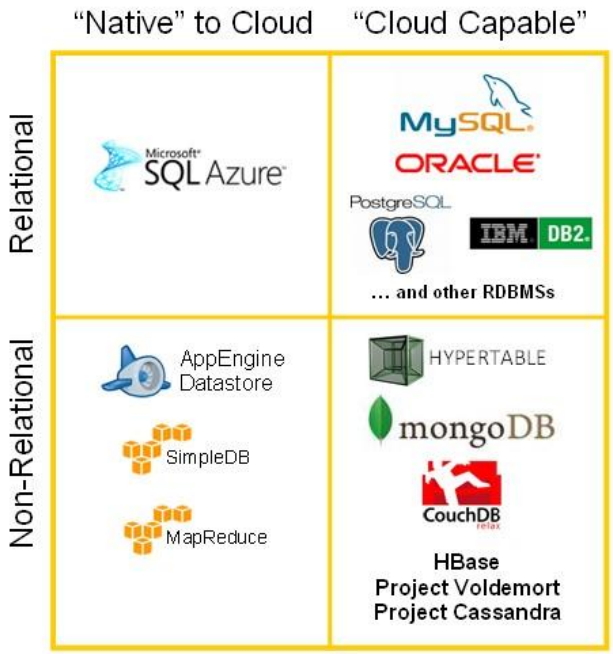

Figure 3 The databases on the cloud [2]

Que faire lorsque l'ont veut déployer différents serveurs d'applications sur plusieurs nœuds EC2 partageant la même base de données ? Quels types d'architecture devrons-nous mettre en œuvre ?

Contribution : L'objectif du mémoire est (1) d'étudier et de décrire les différentes solutions de stockage offertes par Amazon EC2, Oracle et Mysql d'un point de vue architectural, (2) de

déterminer leurs « natures élastiques », (3) de les replacer dans un contexte d'un *cluster* de serveurs d'application, et enfin (4) de proposer une architecture d'exposition de données sur un cluster de nœuds virtuels.

Organisation: ce mémoire est organisé par l'ULB en collaboration avec Euranova R&D. L'étudiant sera accompagné par l'équipe d'Euranova R&D.

CONCEPTION D'UNE ARCHITECTURE DE MIGRATION DE LA PLATFORME DE COMMUNICATION SUR UN CLOUD

Contexte : La plupart des vendeurs de serveurs d'applications propose aujourd'hui une image Amazon afin d'utiliser leurs produits sur un *cloud*. D'autres serveurs d'applications *open source*, pour la plupart, n'ont pas encore été portés sur ce type d'infrastructure. C'est le cas de Mobicents SipServlet, un serveur d'application JAVA http et SIP open source implémenté par Red Hat dans le cadre de sa *Communication Media Platform*. Le but de cette plateforme est de pouvoir créer, déployer, et gérer des services et applications qui intègrent la voix, vidéo et données sur les réseaux IP et de communications.

Eucalyptus est l'une des implémentations de *cloud open source* les plus connues. Eucalyptus compte déjà comme utilisateurs, la NASA, Lilly Pharma et il est à la base de l'*Entreprise Cloud* fourni par la distribution Ubuntu. Le caractère *open source* du *framework* nous permet, non seulement d'étudier l'architecture *cloud*, mais aussi d'expérimenter la migration sur le *cloud* de certains serveurs d'applications à tout niveau de granularité.

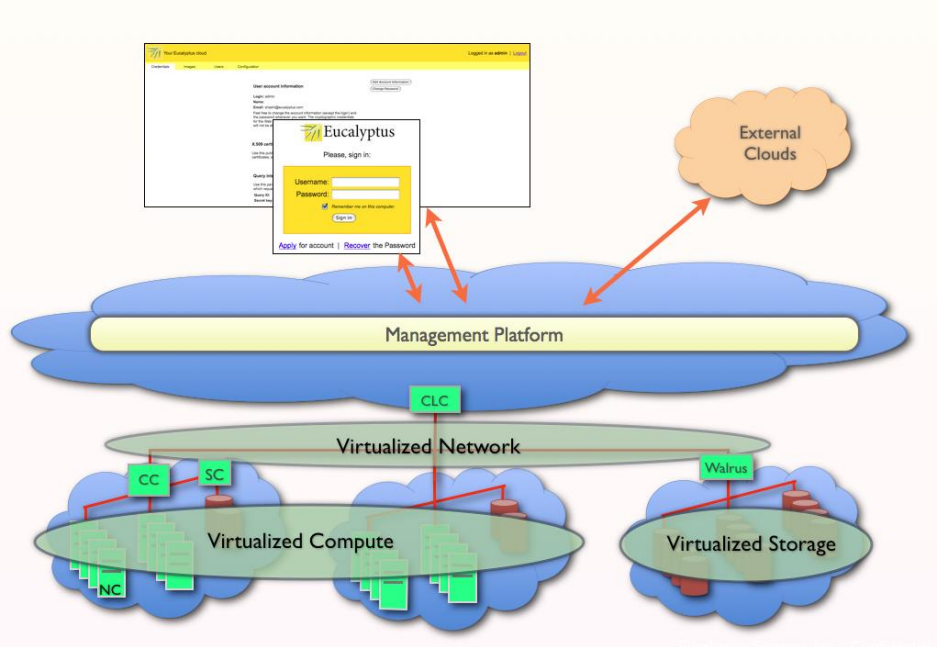

Figure 4 The Eucalyptus cloud and cluster architecture.

Contribution : L'objectif du mémoire est (1) d'étudier et de décrire en détail l'architecture d'Eucalyptus basée sur son *grid*, (2) de proposer l'architecture nécessaire à la migration du SipServlet sur Eucalyptus, en tenant compte des aspects de clusterisation et enfin (3) d'implémenter un prototype d'un serveur SipServlet sur Eucalyptus.

Organisation: ce mémoire est organisé par l'ULB en collaboration avec Euranova R&D. L'étudiant sera accompagné par l'équipe d'Euranova R&D.

ETUDE ET IMPLEMENTATION D'UN ORDONNANCEUR DE JOBS SUR GRID

Contexte : Les grilles de calculs (*grids*) sont aujourd'hui utilisées dans différents domaines (bancaire, pharmaceutique, industrie, biologie moléculaire, physique appliquée, etc.). Pour suivre cette tendance, Red Hat propose le framework MRG (*Messaging, Real-Time, Grid*), dont la partie *grid* est implémentée par le projet Condor, un projet *open source* développé par l'université du Wisconsin [4].

Dans le cadre d'implémentations de projets spécifiques nous avons besoin de localiser certaines données sur des nœuds particuliers du *grid*. C'est typiquement le cas lorsque de grandes quantités de données doivent être utilisées par les *jobs*. Les données sont alors distribuées sur des nœuds particuliers de la grille. Dès lors nous devons étendre la politique de distribution de *jobs* en tenant compte de la localisation de ces données, ce que ne fait pas l'ordonnanceur de Condor.

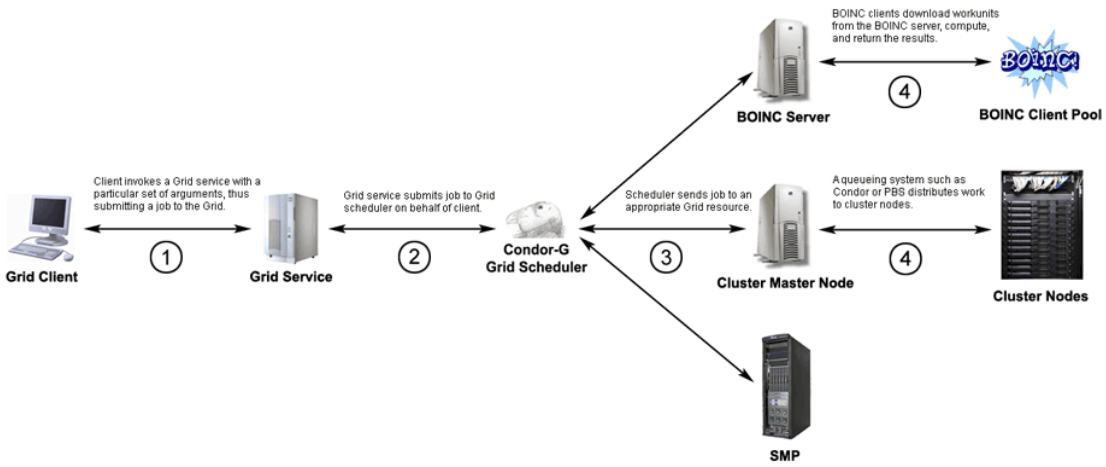

Figure 5 ***BOINC Project: utilisation de Condor [5].***

Ce type d'architecture commence à être implémentée dans les applications à haute disponibilité comme Cassandra DB, la base de données de Facebook ou encore BigTable de Google.

Contribution : L'objectif du mémoire est (1) d'étudier et de décrire l'architecture du projet Condor, (2) d'étudier les politiques de distribution de *jobs* et l''implementation de l'ordonnanceur, (3) étendre l'algorithme de distribution en tenant compte de la localisation des données et enfin (4) de proposer un prototype et de le déployer sur une grille Condor.

Organisation: ce mémoire est organisé par l'ULB en collaboration avec Euranova R&D. L'étudiant sera accompagné par l'équipe d'Euranova R&D.

COMPARAISON DES APPROCHES NOSQL ET VALIDATION D'UNE ARCHITECTURE DE DONNEES A HAUTE CAPACITE

Contexte : NoSQL (Not Only SQL) est un mouvement relatif aux bases de données, entamé au printemps 2009. Le terme se réfère à certaines données non relationnelles stockées. Des tendances dans les architectures informatiques visent à améliorer les bases de données dans une direction nécessitant une montée en charge via une clusterisation linéaire. NoSQL tente de répondre à cette exigence en fournissant un système moins transactionnel mais hautement distribué. Les implémentations les plus connues sont Google BigTable et Amazon Dynamo, mais de nombreux projets Open source commencent à voir le jour, comme Cassandra (FaceBook), Voldemort (LinkedIn), HBase (BigTable), etc.

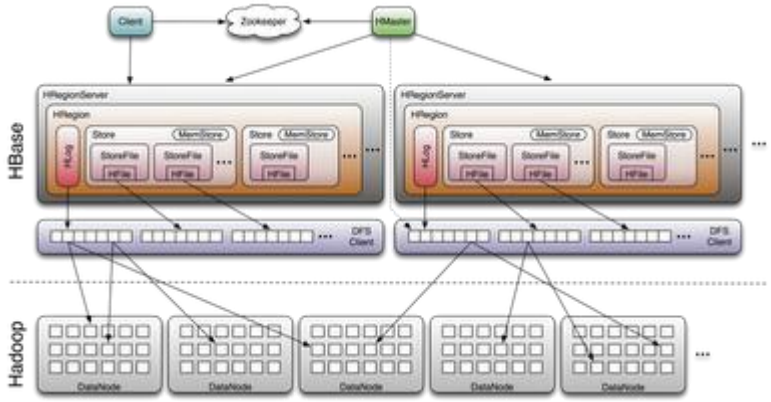

Figure 6 The HBase integration with Apache Hadoop.

Cependant, nous ne pouvons pas remplacer les systèmes de données relationnelles sans étudier les implications que le déploiement d'un Framework NoSQL peut avoir sur les données et l'infrastructure IT. En effet, nous devons au préalable étudier les points suivants :

1. Les modèles de consistance de données,
2. Les modèles de scalabilité en termes de politique d'indexation, de *look-up*, de *sharding*, de routage vers les *sharding*, etc.
3. Les modèles transactionnels,
4. Le modèle de requête,
5. Capacité d'intégrer le système de routage des requêtes en vue de l'intégration d'un serveur d'application distribué.

Contribution : L'objectif du mémoire est (1) d'étudier de d'analyser les projets HBase, Cassandra et Voldemort sur les 5 points cités ci-dessus, (2) de concevoir et d'implémenter un prototype d'application test sur un *framework* NoSQL choisis par l'étudiant et enfin, (3) d'effectuer des tests de charges sur l'application et de valider l'architecture mise en place.

Organisation: ce mémoire est organisé par l'ULB en collaboration avec Euranova R&D. L'étudiant sera accompagné par l'équipe d'Euranova R&D.

RÉFERENCES

[1] http://code.google.com/p/mobicents/

[2] http://www.cloudbzz.com/wp-content/uploads/2009/07/ScreenHunter_0194.jpg

[3] The Eucalyptus web site, http://open.eucalyptus.com

[4] University of Wisconsin, http://www.wisc.edu/

[5] Description du projet BOINC, http://ditwww.epfl.ch/SIC/SA/SPIP/Publications/spip.php?article928

[6] NoSQL Wikipedia Definition, http://en.wikipedia.org/wiki/NoSQL

[7] Van Den Bossche B. and al., *Design of a JAIN SLEE/ESB-based platform for routing medical data in the ICU*, in Computer methods and programs in biomedicine 91 (2008) 265–277 Ghent (BEL) University