



Change the game with smart
innovation

Master Thesis 2013 – 2014 Faculty of Science
engineering

12/08/2012

Master Thesis proposal for the academic year 2013.

TABLE OF CONTENTS

Section Un	Introduction	3
	EURA NOVA R&D	3
Section Deux	Mémoires 2013.....	4

	Evolution des systemes de sockage et convergence des structures relationnelles et nosql : Etude, modelisation et comparaison.	5
	Convergence entre Calcul GPU haute performance et traitement de donn ées distribu é: Conception d'une nouvelle génération de plateforme de calcul	7
	Remplacement d'un système de fichier distribué par un grille de données en mémoire pour un framework de traitement de donn ées distribu ées bas ésur les DFG	9
	AROM Processing Framework – Evolutions et nouvelle generation de traitement de donn ées.	10
	Mise en place d'un outil de création et de manipulation de diagrammes représentant un mod èle métier.	12
	Proposition d'une nouvelle approche pour l'indexation de graphes sociaux.	14

	R éferences	16
--	-------------------	----

SECTION UN

INTRODUCTION

EURA NOVA R&D

EURA NOVA est une société Belge constituée depuis le 1er Septembre 2008. Notre 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. EURA NOVA découple la gestion de carrière de la relation client en s'appuyant notamment

sur une perception entrepreneuriale de la carrière.

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 EURA NOVA.

SECTION DEUX

MÉMOIRES 2013



Massachusetts
Institute of
Technology



EVOLUTION DES SYSTEMES DE STOCKAGE ET CONVERGENCE DES STRUCTURES RELATIONNELLES ET NOSQL : ETUDE, MODELISATION ET COMPARAISON.

Contexte : Le NoSQL est un ensemble de technologies et de techniques développées par le monde internet pour satisfaire d'importants besoins de scalabilité. Leur choix était de sacrifier l'aspect transactionnel au prix d'une exceptionnelle performance en écriture, une distribution efficace des lectures et surtout une élasticité du stockage et ce en respectant deux et trois propriétés du désormais célèbre théorème de CAP. Mais aujourd'hui la situation a changé et le contexte du théorème a évolué [1]. Les acteurs du Web réalisent que même dans des applications sociales, le temps réel et la consistance forte sont devenus des aspects primordiaux. De plus, le NoSQL a vu une importante adoption par la communauté et le besoin d'implémenter les fonctionnalités de consistance au niveau applicatif sont devenus importants. C'est la raison pour laquelle les bases de données NoSQL ont récemment évolué vers des architectures distribuées permettant l'exécution de transactions, typiquement via des mécanismes d'élection de leader. C'est ce qu'El-Abadi décrit comme la *fusion de données* [2] : l'évolution des structures NoSQL vers des fonctionnalités des RDBMS. C'est le cas de Google Megastore [3] ou de G-Store [5].

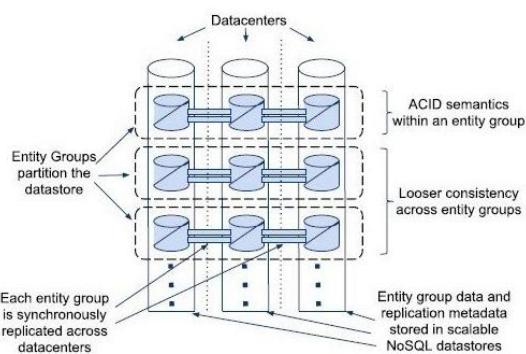


Figure 1 the Google Megastore architecture[3]

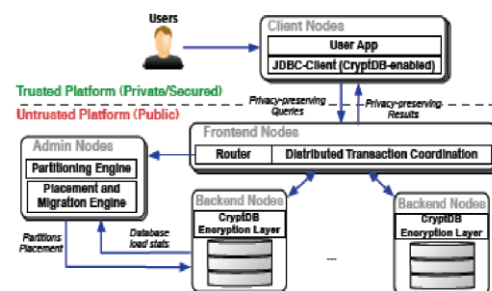


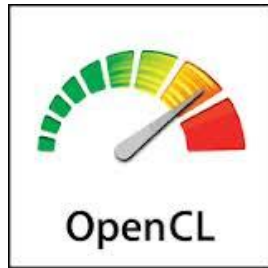
Figure 2 the MIT RelationalCloud architecture [4]

D'un autre côté, sous la pression des NoSQL, les bases de données relationnelles ont évoluées vers des structures plus performantes ou plus élastiques. Les deux plus importantes évolutions sont les bases de données orientées colonnes et les évolutions vers des fonctionnalités NoSQL. Cette dernière est décrite par El-Abadi comme la *fission de données* [2]. C'est le cas de Elastras, ClouddSQL Server de Microsoft research ou encore relationalCloud du MIT.

L'objectif de ce mémoire est donc d'étudier ces deux grandes évolutions, data fission et fusion, d'en analyser et modéliser les architectures, mais d'aussi modéliser leurs comportements élastiques en s'inspirant des travaux d'EURA NOVA en collaboration avec l'ULB et l'UCL en la matière.

Contribution : l'objectif de ce mémoire est (1) d'étudier et de modéliser les nouvelles bases de données en incluant Elastras, RelationalCloud, Spanner, Megastore et G-Store (2) d'en déduire le nature élastique via la définition ou l'adaption de modèles existants et enfin (3) de valider et comparer leurs comportements par une la mise en place d'un benchmark.

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



CONVERGENCE ENTRE CALCUL GPU HAUTE PERFORMANCE ET TRAITEMENT DE DONNÉES DISTRIBUTÉ : CONCEPTION D'UNE NOUVELLE GÉNÉRATION DE PLATEFORME DE CALCUL

Contexte : Le monde du *High Performance Computing* (HPC) est un écosystème principalement composé de chercheurs travaillant sur des problèmes de traitement et d'analyse de grands volumes de données. En fait, ces calculs scientifiques propres à la physique, l'astronomie ou la biologie sont les premières vraies formes de Big Data. Les premières technologies développées dans ce cadre tournaient autour du Grid Computing. Aujourd'hui la communauté HPC voit émerger les calculs utilisant les GPU, les processeurs des cartes graphiques.



Figure 3 les GPU sont devenus une source importante de traitement de données.

Cependant, force est de constater que l'industrie IT a pris une voie tout à fait différente dans le traitement des grands volumes de données. D'abord le monde internet s'est massivement investi dans les frameworks de traitement distribués comme Hadoop MR ou les data flow Graph comme AROM [6] développé par EURA NOVA. Ensuite, le monde de l'entreprise IT, fortement focalisé sur les data Warehouse, a lui évolué vers des solutions *in-memory* en favorisant des *appliances* avec des configurations HW très performante utilisant des connections INFINIBAND ou uSNIC (User Space Network Interface Card) permettant des lectures RDMA (Remote Data Memory Access) de l'ordre de la micro seconde. Les algorithmes de mining étant, pour la plupart implémentés directement *in-DB* et utilisant au maximum le potentiel du HW.

L'objectif de ce mémoire est de valider la pertinence d'une architecture de traitement de données haute performances hybride permettant d'exploiter au maximum les GPU et les configurations

hautes gammes des machines tout en prenant avantage d'une architecture et d'un framework de traitement distribué pour l'exécution d'algorithmes d'analyse de données. L'étudiant prendra un algorithme de data mining, soit en classification ou clusterisation, comme exemple pour valider l'approche. Enfin, l'étudiant proposera une architecture de la solution hybride.

Contribution : l'objectif de ce mémoire est (1) d'étudier les modèles programmatiques et la littérature liés à la programmation HPC GPU, (2) d'étudier les architectures de traitement de données distribués, (3) de proposer et de concevoir une solution qui permet l'ordonnancement d'un traitement de données sur un cluster GPU déployer sur plusieurs machines.

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

REPLACEMENT D' UN SYSTÈME DE FICHIER DISTRIBUÉ PAR UN GRILLE DE DONNÉES EN MÉMOIRE POUR UN FRAMEWORK DE TRAITEMENT DE DONNÉES DISTRIBUÉES BASÉ SUR LES DFG

Contexte : Les caches distribuées sont des extensions du concept de cache s'étendant sur un *cluster* de machines et augmentant ainsi la capacité de transactions sur la cache. Récemment les développements technologiques dans le domaine ont mené à la spécification (JSR 347) d'une nouvelle génération de cache distribuée : les *data grids*. A la différence des caches traditionnelles elles offrent des fonctions d'indexation, de distribution de données, et des API de type SQL et OQL.

Dans le cadre d'un précédent mémoire mené à l'Université Libre de Bruxelles (ULB), un prototype de framework de processing distribué avait été conçu et utilise les DataFlow Graphs pour la définition des jobs à exécuter [6]. Ce modèle de processing est plus général et flexible par rapport à MapReduce et permet ainsi de concevoir des jobs itératifs ou à l'allure de *pipeline* de manière optimale.

A ce stade-ci, le framework supporte le Hadoop Distributed FileSystem (HDFS), faisant partie de la suite d'outils Apache Hadoop, comme système de fichier distribué. Ce système de fichier persiste et réplique les données sur le disque dur, et distribue les *chunks* de données sur les différents nœuds présents dans le cluster.

L'objectif de ce mémoire est de concevoir et implémenter le support d'une data grid comme système de fichiers distribué pour AROM. De plus, la conception de ce support devra tenir compte des différentes stratégies de *caching* et de *flushing* des données propres aux data grids en fonction des données traitées et des types de jobs exécutés par le framework.

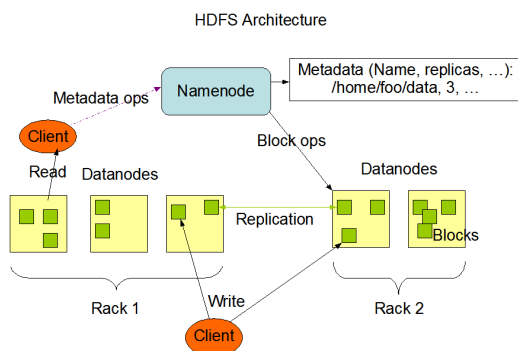


Figure 4 Architecture de HDFS

Contribution : l'objectif de ce mémoire est (1) d'étudier les solutions de data grid existantes, (2) de proposer un modèle d'accès à la data grid pour le framework de processing distribué, (3) d'implémenter un driver permettant à AROM d'utiliser la data grid comme système de fichier selon

le modèle proposé en (2), et (3) de proposer des stratégies de caching et de flushing des données sur la datagrid qui soient efficaces en regard des types de jobs exécutés sur le framework.

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



AROM PROCESSING FRAMEWORK – EVOLUTIONS ET NOUVELLE GENERATION DE TRAITEMENT DE DONNÉES.

Contexte : AROM est un framework de processing distribué ayant pour vocation de fournir un « bac à sable » pour la recherche et l'expérimentation sur les traitements distribués. En effet, son modèle de processing se basant sur les DataFlow Graphs est générique et flexible, permettant de concevoir et exécuter une large gamme de jobs : simples, itératif, pipelines, ... AROM fournit ainsi un compromis clair entre performance, scalabilité et flexibilité

Dans une précédente étude, les tests de performance menés à grande échelle (> 40 noeuds) et sur des larges quantités de données (> 40Gb) ont permis de valider la démarche et les concepts techniques à la base de AROM. Cependant, à grande échelle, les faiblesses de l'implémentation actuelle ont été mises en évidence. Elles concernent deux volets fondamentaux de l'architecture d'AROM : (1) la saturation du backbone de communication asynchrone et (2) le scheduling des éléments des jobs sur les ressources disponibles.

En ce qui concerne le premier point, une première analyse a montré que le problème pouvait être adressé en migrant la stack de communication actuelle (Akka 1.0) vers une stack plus performante et adaptée (par ex : Akka 2.0).

De l'autre côté, concernant le scheduling, la prochaine mouture de Hadoop (0.23, estampillée « dotNext »), framework de processing distribué basé sur MapReduce, visera à découpler ses éléments de contrôle (management) afin de fournir des éléments séparés et découplés. Ainsi, le scheduler (YARN- Yet Another Ressource Negotiator) ne sera plus étroitement lié aux jobs MapReduce et sera découplé des autres éléments de Hadoop. Son architecture de plus sera modulaire avec la possibilité de définir ses propres politiques de scheduling. Partant de ce constat YARN pourrait être un candidat à l'amélioration du scheduling actuel de AROM.

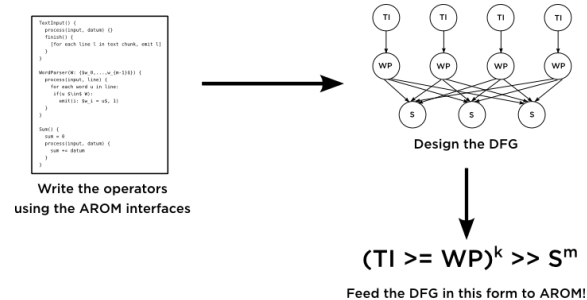


Figure 5 Définition d'un job dans AROM

Contribution : l'objectif de ce mémoire est (1) de proposer un plan de migration d'Akka 1.0 vers Akka 2.0 pour le framework AROM en tenant compte des nouvelles fonctionnalités, (2) d'implémenter la migration et (3) d'étudier YARN, le scheduler de la plateforme Hadoop dotNext et de proposer un modèle d'intégration de ce scheduler dans AROM.

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



MISE EN PLACE D'UN OUTIL DE CRÉATION ET DE MANIPULATION DE DIAGRAMMES REPRÉSENTANT UN MODÈLE MÉTIER.

Contexte : Dans le monde des applications riches on retrouve beaucoup d'outils qui permettent de créer et gérer des diagrammes de toute sorte. Ces diagrammes sont souvent des représentations d'un modèle métier sous-jacent qui contient un ensemble de données qu'on souhaite pouvoir représenter d'une manière ou d'une autre (processus, statistiques, requête, modèle, ...).

Au niveau d'Eclipse les technologies EMF permettent facilement de représenter des modèles de données. Ainsi une librairie telle que Graphiti met à disposition un ensemble de représentations graphiques pour ces modèles. Au delà de la visualisation, il est également possible de créer et de manipuler ces modèles graphiquement. Cette édition graphique permet à des utilisateurs non techniques de manipuler ces modèles.

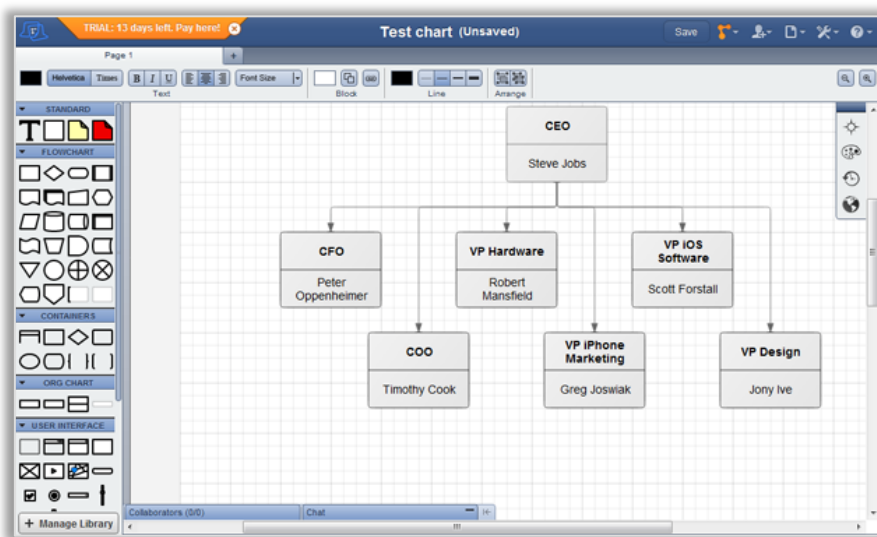


Figure 6 La plupart des framework existant sont soit propriétaire, soit sont orienté solution, comme par exemple une solution BPMN complète, mais aucun n'adresse la problématique d'un modèleur générique.

Malgré l'existence de quelques solutions propriétaires qui ne répondent qu'à des besoins spécifiques, un outil équivalent qui permettrait d'éditer des modèles graphiques directement dans un navigateur web n'existe pas. On y retrouvera néanmoins des librairies pour faire de la visualisation de données tel que D3 ou JIT (Javascript Infoviz Toolkit). On y trouvera aussi des librairies pour faire le lien entre un modèle sous-jacent (potentiellement stocké à distance) et sa visualisation (ex :backbone.js). Il n'existe cependant pas d'outil complet permettant de faire de l'édition de modèles en temps réel.

Un tel outil permettrait à des utilisateurs non-techniques de collaborer à l'élaboration de modèles en temps réel sans rien avoir à installer sur leur poste de travail.

Contribution : Le mémoire sera organisé en deux parties :

1. La première partie consiste à étudier les différentes technologies de représentations des données, ainsi que les solutions d'environnement de développement web (Orion, Cloud9, ...)
2. La deuxième partie consistera à mettre en place et à implémenter avec les technologies les plus pertinentes un outil permettant la visualisation d'un modèle stocké à distance et sa manipulation en temps réel dans un navigateur.

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



PROPOSITION D'UNE NOUVELLE APPROCHE POUR L'INDEXATION DE GRAPHS SOCIAUX.

Contexte: L'exploitation de nouvelles générations de sources de données sociales (e.g. Facebook, Twitter ...), le premier challenge à relever est de trouver une structure de représentation de ces données. Ces données sont généralement hétérogènes (numérique, texte, image, interaction), dynamiques (évolutives) et qui peuvent être liées entre elles. De plus, une telle structure de représentation doit fournir des moyens simples pour exploiter et découvrir des nouvelles connaissances. L'utilisation de graphes comme structure de représentation constitue une solution bien adaptée à cette problématique. Les graphes fournissent une représentation structurée caractérisée par une grande capacité représentative par rapport aux représentations usuelles (i.e. vecteurs, tuples, ensembles...). En effet, les graphes offrent une riche représentation expressive à travers la description explicite de la structure des données, par l'ensemble de ses sous-structures et les interactions entre elles.

Par ailleurs, l'utilisation de graphe, comme modèle de structuration des réseaux sociaux, nécessite tout un ensemble d'algorithmes adaptés pour gérer le stockage et l'accès aux données d'une manière efficace. Ainsi, le développement des algorithmes d'indexation de graphes devient un besoin réel. Des tels algorithmes doivent être conçus en tenant compte de trois aspects cruciaux :

- **L'aspect structurel** : à la différence des moteurs d'indexation classique, qui se focalisent que sur le contenu de données, ces nouveaux algorithmes (indexation et recherche) doivent considérer aussi les relations et les liens qui connectent ces données dans le graphe. Il faut noter ici que l'utilisation des techniques classiques pour indexer un graphe entraîne la perte de l'information structurelle de données.
- **L'aspect dynamique** : la dynamique constitue une des principales caractéristiques des données sur internet. Ainsi, les algorithmes d'indexation doivent prendre en compte cet aspect évolutif en considération en minimisant le temps nécessaires entre la collecte et la mise à disposition (à travers le moteur de recherche) de données.
- **L'aspect passage à l'échelle** : dans le contexte de réseaux sociaux et du contenu généré par les utilisateurs (User generated content), les données forment généralement de larges volumes. Ce qui constitue un nouveau challenge à considérer en concevant ces algorithmes.

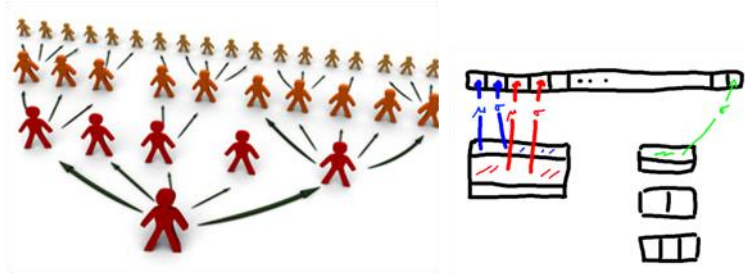


Figure 7 Actuellement, il ya besoin important de trouver de nouvelles approches pour accéder aux données dites sociales

Contribution : L'objectif de ce mémoire est (1) d'étudier les différentes solutions d'indexation de graphe en général tout en se focalisant sur l'indexation de l'accessibilité (« reachability ») (2) d'adapter une solution optimale pour les graphes sociaux (3) et finalement implémenter la (ou les) solution(s) dans la base de données graphes développée au sein d'EURA NOVA.

RÉFÉRENCES

- [1] E. Brewer, *CAP twelve years later: How the "rules" have changed*, in IEEE Computer Journal vol. 45, 2012.
- [2] D. Agrawal and al., *Database Scalability, Elasticity, and Autonomy in the Cloud*, in the Proceedings of the 16th international conference on Database systems for advanced applications - Volume Part I. 2011
- [3] J. Baker and al., *Megastore: Providing Scalable, Highly Available Storage for Interactive Services*, in Proceedings of the Conference on Innovative Data system Research (CIDR) (2011), pp. 223-234. 2011
- [4] C. Curino and al., *Relational Cloud: a Database Service for the cloud*, in CIDR 2011.
- [5] S. Das and al., *G-Store: a scalable data store for transactional multi key access in the cloud*, in the Proceedings of the 1st ACM symposium on Cloud computing. 2010
- [6] NL. Tran and al., *AROM: Processing Big Data With Data Flow Graphs and Functional Programming*, in the CRC workshop of the 4th IEEE International conference on Cloud Computing technology and Science. 2012