

Introduction

1 Contexte de la thèse

Cette thèse est le fruit d'une collaboration entre la société Lokad et le laboratoire Bilab de Telecom ParisTech.

Lokad est une jeune société éditrice de logiciels, spécialisée dans la prévision statistique de séries temporelles. Ces prévisions sont vendues à des entreprises pour gérer et optimiser leurs flux de clients, d'appels ou de stocks. Les clients principaux de Lokad sont les acteurs de la grande distribution, les e-commerce, ... Pour rendre facilement exploitables ces prévisions, Lokad propose des logiciels en mode Software as a Service (SaaS), c'est à dire des applications hébergées par ses soins et accessibles directement depuis internet, qui utilisent ces prévisions pour fournir des outils d'aide à la décision. Lokad fournit par exemple via son application SalesCast un système de gestion des stocks en déterminant les niveaux de stock à maintenir et les réapprovisionnements à effectuer. Lokad propose également via son application ShelfCheck un outil de détection d'indisponibilité d'un produit en rayonnage, reposant lui aussi sur des prévisions de ventes de produits.

Deux caractéristiques de Lokad inscrivent cette société dans les problématiques de calculs intensifs et donc parallèles.

La première de ces caractéristiques est le caractère SaaS des logiciels développés par Lokad. Une utilisation classique des services de Lokad se déroule de la manière suivante: une application SaaS se charge d'accéder à la base de données du client; cette application envoie les données via le réseau sur les serveurs de Lokad; le moteur interne de prévision statistique s'applique aux données pour fournir des prévisions; les prévisions sont retournées au client via le réseau. Cette approche permet à Lokad d'avoir un contrôle fin sur la qualité des prévisions qu'elle délivre, mais elle implique que l'intégralité des tâches de calculs soit assumée par les serveurs de Lokad.

La seconde de ces caractéristiques est la gestion automatisée des prévisions. Lokad a développé un moteur interne de prévisions de séries temporelles qui ne requiert pas de travail direct d'un statisticien: pour chaque série temporelle à prévoir, ce moteur évalue la précision de chacun des modèles disponibles en production par cross-validation, et sélectionne automatiquement les prévisions du modèle le plus précis pour la série. L'automatisation des prévisions permet à Lokad d'adresser la demande de certaines entreprises dont la taille des données rendait très difficile des prévisions plus manuelles. C'est le cas par exemple de la grande distribution, où chaque enseigne possède des centaines de points de vente, et où le nombre de produits par point de vente peut atteindre 50 000 références. Cette automati-

sation implique également que la quantité de calcul à fournir est proportionnelle aux nombres de séries temporelles à prévoir, mais aussi au nombre de modèle et aux nombres de points de cross-validation par modèle. Ainsi l'automatisation vient au prix de calculs bien plus intensifs.

Parallèlement, Telecom ParisTech est un acteur européen de référence dans les domaines des sciences et technologies de l'information et de la communication (STIC). Son laboratoire BiLab, qui travaille en partenariat avec EDF, est spécialisé dans les technologies de l'aide à la décision, plus connues sous le nom de Business Intelligence (BI). En particulier, le Bilab s'intéresse aux problématiques de prévisions statistiques d'EDF. En raison de la taille d'EDF, ces problématiques à l'origine statistiques deviennent aussi liées à l'informatique et à la nécessité de gérer des gros volumes de données et des calculs coûteux.

Telecom ParisTech et Lokad se trouvent donc tous les deux à l'interface entre informatique et statistique, à l'endroit même où les deux disciplines se rejoignent dans la problématique d'algorithmes statistiques répartis.

Le concept récent de Cloud Computing (francisé en "informatique dans les nuages") est en train de modifier profondément le monde informatique. Le Cloud Computing propose une translation depuis le modèle économique dans lequel l'utilisateur possède les logiciels et les infrastructures hardware vers un modèle dans lequel l'utilisateur est un simple locataire de services, qu'ils soient software ou hardware. Le Cloud Computing abstrait et dématérialise l'espace physique dans lequel les calculs et les données sont gérés et ouvre des perspectives nouvelles pour les entreprises mais aussi pour la recherche académique.

Mon travail de thèse s'inscrit au confluent de ces différents sujets: statistique, calcul réparti et Cloud Computing. Mes activités de recherche ont porté sur cette technologie naissante pour laquelle peu de travaux académiques ont à ce jour été réalisés. Mon travail a consisté à explorer les capacités et limites de ces technologies encore mal comprises, pour en appréhender les enjeux et proposer des manières pertinentes de paralléliser des algorithmes. Ce travail d'exploration et de synthèse donne naissance aux chapitres ?? et ??.

Ces algorithmes que j'ai portés sur le cloud se répartissent en deux catégories. La première catégorie regroupe différents composants du moteur de prévision interne de Lokad que j'ai portés sur le cloud. Ces travaux, moins académiques, sont d'autant plus difficiles à partager que le code qui en découle est propriétaire. Ils m'ont cependant permis par induction de dégager quelques considérations générales sur la conception d'application sur le cloud. J'en présente certaines

dans le chapitre ??.

La seconde catégorie d'algorithmes a trait aux méthodes de clustering réparti, qui font l'objet principal des chapitres ?? à ?? du présent document. La problématique de clustering est un aspect très important de la prévision de séries temporelles chez Lokad. En effet, pour améliorer ses prévisions, Lokad utilise des méthodes dites "multi-séries". L'objet de ces méthodes est d'utiliser l'information contenue dans toutes les séries afin d'affiner la prévision de chacune d'entre elles. Un exemple d'utilisation est l'exploitation des saisonnalités : en dégagant des comportements saisonniers d'un ensemble de séries temporelles, certains modèles de Lokad sont capables d'affiner la composante saisonnière, plus bruitée sur une seule série temporelle. Le clustering est donc un outil essentiel pour Lokad, outil qu'il était nécessaire d'adapter à la plateforme de Cloud Computing.

Les travaux scientifiques portant sur des calculs répartis de grande ampleur sont appliqués à des jeux de données dont l'échelle de taille peut se révéler très variable. Pour chaque ordre de grandeur du problème, les outils hardware et software adéquats sont très différents; on conçoit aisément qu'à des problèmes aussi différents en taille que l'indexation du web par Google Search ou un petit jeu en ligne, il est fourni des solutions très distinctes. Il est ainsi primordial de préciser quels sont les ordres de grandeur pour lesquels un travail donné est conçu. Mes travaux de thèses portent sur des calculs dont l'ampleur correspondrait sur une seule machine à une centaine d'heures, avec à ma disposition 250 machines. Les jeux de données les plus gros sur lesquels j'ai fait tourner mes algorithmes représentent environ 50 Go. J'ai choisi de concentrer mes travaux sur ces ordres de grandeur qui reflètent la taille des plus gros clients actuels de Lokad.

La suite de cette partie introductive présente succinctement le contenu de chaque chapitre de ce manuscrit.

2 Présentation des travaux

2.1 Chapitre 2 - Introduction au Cloud Computing

Les progrès fulgurants des mécanismes de collecte et de stockage des données n'ont pas été accompagnés de progrès aussi rapides dans le développement des processeurs. Cette réalité a incité au développement de systèmes physiques et logiciels permettant de répartir la charge de travail sur de multiples unités de calcul. Ces calculs intensifs ont tout d'abord été portés sur des architectures physiques dédiées, communément appelées super-calculateurs. Ces super-calculateurs

étaient des dispositifs physiques conçus en faible quantité, spécifiquement pour réaliser des calculs intensifs. L'explosion du marché des ordinateurs personnels dans les années 80, puis d'internet à la fin des années 90 a ouvert des perspectives nouvelles quant à la manière de répartir des calculs. Des systèmes nouveaux sont alors apparus, reposant sur la collaboration de plusieurs agents administrativement distincts et mettant en commun une partie de leurs ressources. Parmi ces systèmes, on trouve des infrastructures de Grid Computing, comme Condor ([?]), ou des systèmes plus récents, par exemple de Peer-To-Peer, comme Napster ou Folding@Home.

La parallélisation de calculs sur une vaste quantité de machines soulève de nombreuses difficultés, qu'elles aient trait à la communication entre les machines, à l'accès en écriture ou lecture à une mémoire partagée efficace, ou à la répartition de la charge de calcul sur les différentes unités disponibles. Les années 2000 ont vu l'émergence d'applications web consommant des ressources immenses : c'est le cas par exemple de Google Search, de Bing, de Facebook, de Youtube, ... Les entreprises à la tête de ces applications ont développé des cadres logiciels (frameworks) mais aussi physiques (via la construction de centre de calculs spécifiques(ou datacenters)) pour proposer des solutions aux difficultés sus-nommées.

Certaines de ces entreprises, rejointes par d'autres acteurs économiques, ont alors proposé de mettre en location un accès à ces différentes solutions, physiques et logicielles. Ces sociétés sont alors devenues des fournisseurs de Cloud Computing.

Le cloud computing a de très nombreuses implications techniques, scientifiques ou commerciales. Les différentes offres se catégorisent selon le fait qu'elles offrent des solutions clef en main, ou qu'au contraire elles laissent plus de libertés à leurs utilisateurs au prix d'une complexité d'utilisation plus forte. Parmi les solutions clef en main, on recense les offres de Software as a Service (SaaS), comme Google Search, Gmail ou Facebook. On y regroupe également de nombreuses solutions spécifiques, développées pour chaque client par des sociétés tierces (notamment des sociétés de services en ingénierie informatique (SSII)). Ce chapitre présente principalement les catégories d'offres de cloud computing les plus à même de permettre l'usage de calculs intensifs. Ces offres, souvent plus standardisées et orientées performances, s'adressent à un public de développeurs exclusivement. Parmi elles, nous approfondirons les différences entre les offres de Platform as a Service (PaaS) et celles d'Infrastructure as a Service (IaaS).

Les offres de cloud computing analysées proposent un ensemble de technologies -parfois pré-existantes au concept de cloud computing- réunies dans une

même pile technologique. Le chapitre ?? traite de ces différentes technologies. En particulier, nous donnons un aperçu des difficultés que soulèvent un système de stockage de données réparti sur un ensemble de machines distantes. Nous présentons également certains des environnements logiciels construits pour aider le développement d'applications réparties, comme le célèbre MapReduce originellement développé par Google.

+ PRÉSENTATION DU CHAPITRE

2.2 Chapitre 3 - Introduction à Azure

Microsoft est une société à la fois fournisseur et consommateur de cloud computing : consommateur tout d'abord en raison de ses besoins en interne, notamment pour le moteur de recherche Bing ou pour le stockage de certaines données comme celles liées à la plateforme de jeu en ligne de la XBox 360. Ces besoins en interne ont amené Microsoft à fournir des efforts importants de recherche et développement dans les infrastructures et le logiciel pour disposer d'une plateforme puissante. Ces efforts ont abouti sur des composants comme Cosmos, Scope ([?]), Dryad ([?]) ou DryadLINQ ([?]).

A la suite d'Amazon ou de Google, Microsoft s'est lancé dans l'offre de cloud pour devenir également fournisseur de cloud computing. Son offre se répartit en deux grandes familles de produit. La première regroupe les applications classiques de Microsoft auparavant proposées en mode "desktop" et aujourd'hui disponibles en version cloud. Parmi elles, on peut citer Office 365, qui est la version SaaS du pack Office. Les offres de SQL Server ont également été portées sur le cloud pour prendre le nom de SQL Azure. Cette migration sur le cloud ne se fait pas sans certaines difficultés liées au contexte de l'activité de Microsoft : les offres SaaS de Microsoft font en quelque sorte concurrence à leur pendant desktop, mais elles modifient également le positionnement de Microsoft par rapport à certains de ses partenaires dont l'activité principale consistait à garantir la bonne installation et maintenance de ces applications desktop chez les clients de Microsoft. Cependant Microsoft a affiché sa volonté de développer ces offres qui seront très probablement une part importante de son activité dans les années prochaines.

Enfin, Microsoft propose une plateforme de cloud computing destinée aux éditeurs logiciels, appelée Azure. Cette plateforme permet à ses clients d'héberger des applications en SaaS, qui utilisent les ressources de CPU, de stockage et de bande-passante d'Azure. Microsoft a réutilisé une partie de la technologie développée dans Cosmos (comme détaillé dans [?]) pour construire un système de stockage dédié pour cette plateforme, appelé Azure Storage.

Le chapitre ?? présente les caractéristiques et les enjeux techniques de cette plateforme. En particulier, nous nous intéressons aux abstractions de stockage proposées, et comment elles se sont positionnées par rapport aux différents compromis nécessaires des solutions de stockages réparties, comme introduit dans le chapitre ?. Nous présentons également quelques clefs du développement d'applications sur cette plateforme.

+ PRÉSENTATION DU CHAPITRE

2.3 Chapitre 4 - Un bref aperçu de mon travail au sein de Lokad

2.4 Chapitre 5 - Algorithmes de K-Means répartis

Les algorithmes de clustering ont un rôle central en statistique. D'un point de vue théorique, ils représentent un problème d'apprentissage non supervisé très étudié. D'un point de vue pratique, ils sont un outil souvent indispensable pour l'exploration des données. Ces algorithmes proposent de résumer un jeu de données, réputé d'un certain volume, en un jeu de données plus petit mais cependant représentatif des données initiales. Ce résumé est réalisé en constituant des sous-groupes (également appelés clusters), déduits du jeu initial par le regroupement des données les plus proches suivant un certain critère de similarité.

Etant donné un critère de similarité et un nombre fixé de clusters, le problème de trouver un regroupement optimal au sens du critère est un problème calculatoirement très difficile, souvent irréalisable dans la pratique, dès que la taille du jeu de données dépasse la centaine de points. Des algorithmes d'approximation ont été proposés, fournissant une solution proche de la solution optimale. Parmi ces algorithmes, le Batch K-Means est un algorithme populaire, connu notamment pour sa simplicité de mise en oeuvre.

Les algorithmes de clustering comme le Batch K-Means sont utilisés dans des domaines très variés, qu'il s'agisse par exemple de biologie moléculaire, d'analyse de séries temporelles ou d'indexation du web pour les moteurs de recherche... Chez Lokad, ces procédures sont utilisées pour former des sous-groupes au sein desquels les séries temporelles partagent un même comportement saisonnier. Il est alors plus aisé pour Lokad d'extraire une composante saisonnière de chaque cluster plutôt que de chaque série temporelle, car la composante saisonnière, moyennée sur le groupe de séries, se révèle souvent moins "bruitée" que lorsqu'elle est

extraite série par série.

Les algorithmes de clustering ont récemment connu le développement d'un nouveau défi: celui de la parallélisation. Depuis quinze ans, les moyens de stockage des données (disque-dur, mémoire RAM, etc...) ainsi que des outils de collecte de données ont progressé à un rythme bien supérieur à celui des processeurs. Cette évolution à deux vitesses a conduit les statisticiens à paralléliser leurs tâches de calcul sur de nombreuses machines, mêlant ainsi toujours plus statistique et informatique.

Les travaux du chapitre ?? portent sur le problème du K-Means réparti, en particulier leur adaptation aux nouvelles architectures de calcul sur le cloud, développées au chapitre ?? et ??. Plus précisément, ils montrent comment le premier cadre théorique de K-Means réparti, proposé par Dhillon et Modha ([?]) dans le cas d'une architecture SMP disposant d'un framework MPI ne s'appliquent que partiellement au cloud. Ce chapitre propose une modification de l'algorithme qui permette de l'adapter à l'infrastructure proposée par Azure.

+ PRÉSENTATION DU CHAPITRE

2.5 Chapitre 6 - Considérations pratiques pour les algorithmes de Vector Quantization répartis

Ce chapitre ainsi que le suivant présentent des travaux réalisés et écrits en collaboration avec Benoit Patra, qui était également doctorant au sein de la société Lokad. Nous sommes partis du constat du chapitre précédent selon lequel les communications sur une plateforme de Cloud Computing via le storage sont coûteuses. La distribution du calcul d'un Batch K-Means, bien qu'assez satisfaisante, y est donc ralentie par les nombreuses communications nécessaires ainsi que par les processus de synchronisation de toutes les machines. Nous avons donc travaillé à la parallélisation d'un autre algorithme de clustering, connu sous le nom de Competitive Learning Vector Quantization (CLVQ) qui supprime ou limite ces problèmes et se montre donc selon nous plus adapté à une implémentation sur le cloud.

Tout comme le Batch-KMeans, l'algorithme de CLVQ permet de calculer des clusters pertinents en affinant des points de référence appelés centroïdes ou prototypes qui représentent les différents clusters. Bottou et Bengio remarquent dans [?] que l'algorithme de CLVQ peut être vu comme la version "en-ligne" du Batch-KMeans: au fur et à mesure que le CLVQ traite des points tirés des données à

résumer, il fait évoluer en conséquence les prototypes plutôt que de faire évoluer les points une fois qu'il a examiné toutes les données comme c'est le cas dans le Batch-KMeans.

D'un point de vue statistique, le passage à un algorithme "en-ligne" a présenté des difficultés particulièrement intéressantes. En effet, ce type d'algorithmes est par nature séquentiel, et la parallélisation du calcul est moins naturelle que dans le cas d'un Batch-KMeans, pour lequel une même instruction doit être appliquée indépendamment à de nombreux points (on parle alors de tâches SIMD (Single Instruction, Multiple Data)). Par certains aspects, le CLVQ appartient à la grande famille des algorithmes de descente de gradient stochastique. La parallélisation de ce genre d'algorithmes a déjà fait l'objet d'études, par exemple dans [?], [?] ou [?]. Dans le cas où la fonction objectif est convexe et suffisamment régulière, les stratégies classiques de parallélisation d'algorithmes de descente de gradient par moyennage des résultats des différentes machines mènent asymptotiquement à des speed-ups optimaux, comme démontré dans [?], et obtiennent dans la pratique des résultats très satisfaisants, comme dans [?].

Dans le cas du CLVQ, la fonction objectif n'est ni convexe ni suffisamment régulière, et le moyennage des résultats ne mène pas dans le cas général à de meilleures performances que l'algorithme séquentiel exécuté sur une seule machine. Ce résultat, surprenant, nous amène à reformuler l'algorithme proposé sous une forme qui souligne l'importance de la vitesse de décroissance du pas associé à la descente de gradient. Ce travail nous permet de donner un éclairage différent au précédent algorithme pour appréhender les mécanismes qui l'empêchent d'obtenir des speed-ups satisfaisants.

Enfin, nous proposons un nouvel algorithme de CLVQ réparti, qui évite les travers du précédent. En particulier, notre algorithme va additionner les termes de descente provenant des différentes machines, plutôt que d'en moyennner les valeurs. Cette modification de l'algorithme permet d'obtenir de bons speed-ups présentés en fin de chapitre.

+ PRÉSENTATION DU CHAPITRE

2.6 Chapitre 7 - Le projet CloudDALVQ

Dans ce dernier chapitre, nous présentons le projet logiciel Cloud-DALVQ. Ce projet est la réalisation des algorithmes étudiés dans le chapitre précédent sur la plateforme de Cloud Computing de Microsoft : Windows Azure. Des implémentations ambitieuses d'algorithmes répartis de descente de gradient stochastique

ont déjà été proposées, comme c'est le cas dans [?], [?] ou [?]. Cependant, ces travaux se sont à notre connaissance restreint au cas plus simple où l'infrastructure de calcul possède une mémoire partagée efficace et où les coûts de communication sont donc très faibles ([?], [?]), ou à un cas où la convexité de la fonction objectif permet de supprimer les communications inter-machines sauf à la fin de l'algorithme ([?]). Ainsi, aucun des travaux précédents ne proposait d'exemples d'implémentation d'algorithmes de descente de gradient répartis dans un contexte où les communications sont nécessaires et coûteuses.

D'un point de vue informatique, l'aspect "en-ligne" de l'algorithme de CLVQ nous a semblé particulièrement pertinent, puisque c'est la suppression du caractère batch qui permet de retirer la nécessité de la synchronisation des machines. Le passage vers un algorithme "en-ligne" parallélisé nous permet donc de construire un algorithme asynchrone, libérant nos machines de processus de synchronisation pénalisant les performances. Ce passage vers l'asynchrone permet dans les faits de se libérer de deux mécanismes de synchronisation : le premier a trait aux organisations inter-machines, puisque chaque machine n'est plus bloquée dans ses communications et dans ses calculs par l'attente des résultats des autres unités de calcul. Le second mécanisme que nous supprimons est celui de la séquentialité lecture/calcul/écriture à l'intérieur de chaque machine. Par la suite, nous faisons référence à cet algorithme réparti asynchrone sous l'acronyme de Distributed Asynchronous Learning Vector Quantization (DALVQ). Dans le cadre du DALVQ, chaque machine peut effectuer en parallèle (par un multithread) les tâches de calcul et de communication pour exploiter au mieux les ressources de CPU et de bande-passante.

+ PRÉSENTATION DU CHAPITRE