PLAN DE COURS
Automne 2021

UQAC

8INF803 • Bases de données réparties (Groupe 1)

Enseignant:	Edmond La Chance
Bureau:	P2-4170
Courriel:	Edmond_Lachance@uqac.ca
Page web du cours:	https://github.com/edmondlachance/8INF803

Contenu général

De nos jours, de nombreux sites/applications ont des millions d'utilisateurs. Ces derniers contribuent continuellement de l'information. Toute cette information, ce data, doit être enregistré et traité afin de fournir un service aux utilisateurs. Quelques exemples: Google (engin de recherche), Netflix (suggestions), Instagram (images, comments, votes), Facebook (photos, posts, likes, suggestions d'amis).

Afin d'obtenir un système capable de supporter ce "Big Data", il est nécessaire d'avoir une bonne architecture, et d'utiliser des technologies appropriées. La première nécessité de notre application est le stockage des données. Il faut pouvoir répartir les données sur plusieurs machines, car les quantités de données sont énormes. Nous allons voir deux approches différentes pour répondre à ce problème: l'approche dite incrémentale, et l'approche par précomputation. L'approche incrémentale envoie les données dans un receptacle structuré, ce qui permet de faire des requêtes de lecture plus rapidement par la suite, alors que l'approche par précomputation envoie les données directement dans un log, ce qui est extrêmement rapide pour l'écriture mais vient avec ses désavantages.

La deuxième nécessité est de pouvoir faire des requêtes sur les données afin de les transformer et les traiter. Nous allons utiliser des requêtes de style MapReduce et SQL pour répondre à cette nécessité. Nous verrons également comment exécuter des algorithmes itératifs plus complexes, et comment optimiser la performance de ceux-ci.

La troisième nécessité est le temps de réponse; personne ne veut utiliser une application sociale qui intègre ses données récentes avec trois heures de retard. Nous allons voir les techniques et stratégies qui peuvent être employées pour répondre à cette nécessité.

Le cours a donc comme objectif de montrer à l'étudiant plusieurs facettes du Big Data. Au départ, nous regarderons l'architecture Lambda, une architecture qui propose des stratégies et des technologies pour contrer certains problèmes courants qui surviennent lorsque nous traitons des quantités massives de données. Pour ce qui est du traitement des données, nous allons utiliser le framework Apache Spark, un framework open-source qui est activement utilisé dans l'industrie.

Objectifs du cours

Au terme de ce cours, l'étudiant aura rafraichi et acquis les compétences suivantes :

- Retour sur la BD relationnelle centralisée.
- Normalisation (1FN, 2FN, 3FN)

- Fragmentation horizontale et verticale
- Les connaissances enseignées par l'architecture Lambda
- Des algorithmes utilisés en bases de données réparties : Hyperloglog, Log Structured Merge Tree et Bloom Filter.
- Utilisation du langage SQL
- Programmation MapReduce
- Programmation avec Scala
- Programmation avec Apache Spark
- Connaissance approfondie de Apache Spark, et de son fonctionnement interne.
- Programmation d'algorithmes MapReduce itératifs
- Programmation d'un robot d'indexation et utilisation des expressions régulières

Sujets abordés

Le cours se divise en deux grandes parties.

1. Dans la première partie du cours, nous faisons un retour sur les approches de bases de données traditionnelles (Modèle relationnel et normalisation) et nous parlons du problème de la scalabilité, la capacité d'un système informatique à répondre à une demande en temps de calcul/stockage toujours plus grande. L'Architecture Lambda, une architecture pour faire des sites internets et applications Big Data, est ensuite introduite. Cette architecture recommande d'utiliser plusieurs technologies différentes, afin de satisfaire les nombreux besoins d'un système Big Data comme la vitesse des requêtes, la scalabilité, la consistence des données et la disponibilité des données.
2. La deuxième partie du cours porte sur la programmation avec Apache Spark, un framework populaire de cluster-computing. Tout comme une bonne connaissance du langage SQL permet de traiter les données d'une base de données relationnelle, une bonne connaissance de la technologie Spark permet de traiter les données d'un log réparti ou d'un stream très efficacement. Apache Spark peut également être utilisé pour programmer des algorithmes distribués. Nous montrons également à l'étudiant comment optimiser ses algorithmes en explicant le fonctionnement interne de cette technologie.

   Pendant les cours, le professeur donne un cours magistral aux étudiants fait de théorie et d'exemples de programmation. Par la suite, pendant le restant de la période, il aide les étudiants avec l'installation des technologies, ainsi qu'avec toutes leurs questions. Les étudiants doivent se mettre en équipe et travailler sur deux travaux maisons. Ce sont ces travaux qui sont évalués à la fin du cours.

Évaluation

Deux travaux à faire (1 à 4 personnes) à remettre les **28 Octobre 2020** et **16 Décembre 2021**. Les travaux seront composés de questions à développement ainsi que d'exercices de programmation faisant appel aux technologies et aux concepts présentés en classe. Tout travail remis en retard sans motif valable sera pénalisé de 10% par jour de retard.

Le premier travail compte pour **30%** de la note, et le deuxième travail compte pour **50%** de la note. Il y a également un examen à la fin du cours qui compte pour **20%** de la note finale, la date de cet examen est le **16 Décembre 2021**.

La note finale, sur 100, est la somme de toutes ces évaluations. La note de passage est fixée à **60%** ou D. La date de la fin de la session est le 21 Décembre 2020.

Date limite d'abandon sans mention d'échec

20% de l'évaluation aura été transmise à l'étudiant avant la date limite d'abandon sans mention d'échec, soit le 8 novembre 2021.

Qualité du français écrit

Tout travail remis doit être conforme aux exigences de la politique institutionnelle en matière de maitrise du français écrit. Comme il en est fait mention dans le Manuel de gestion (3.1.1-012).[1] Tout travail dont la qualité du français serait jugée non conforme par l'enseignant pourra être pénalisé jusqu'à concurrence de 10% du résultat maximal prévu.

Évaluation de la qualité de l'enseignement

Ce cours sera évalué en fonction de la Procédure relative à l'évaluation des activités aux programmes d'études de cycles supérieurs (Manuel de gestion, 3.1.2-008).

Formule pédagogique

Le cours sera dispensé en classe par un professeur. Il n'y aura pas de séance de travaux pratiques associées à ce cours.

Situation du cours dans le programme

Le cours est optionnel pour les étudiants inscrits aux programmes de cycles supérieurs. Aucun cours ne lui est préalable.

[1] http://www.uqac.ca/direction_services/secretariat_general/manuel/index.pdf

Références

Aucun livre ou recueil de notes n'est obligatoire pour ce cours. Les diapositives utilisées durant les séances et les lectures préalables seront rendues disponibles en format électronique sur le site du cours.

Ces livres ont été utilisés pour construire le cours :

- Nathan Marz (2015). Big Data: Principles and best practices of scalable realtime data systems
- Michael S. Malak and Robin East (2016). Spark GraphX in Action
- Petar Zečević and Marko Bonaći (2016). Spark in Action
- Holden Karau, Rachel Warren (2017). High Performance Spark