PLAN DE COURS
Automne 2018

UQAC

8INF803 • Bases de données réparties (Gr 1

Enseignant:	Edmond La Chance
Bureau:	P2-4170
Courriel:	edmond.lachance@gmail.com
Page web du cours:	https://github.com/edmondlachance/8INF803

Contenu général

Nous vivons à l'époque des réseaux sociaux. Les réseaux sociaux connectent les utilisateurs d'internet ensemble et leurs fournissent un service avec les données que ces derniers produisent. De nombreux sites internet et applications mobiles ont des millions d'utilisateurs. Ces utilisateurs génèrent une grande quantité d'information tous les jours. Toute cette information, ce data, doit être enregistré et traité et/ou indexé afin de fournir un service aux utilisateurs. Exemples de sites et applications Big Data : Google (engin de recherche), Netflix (suggestions de films), Instagram (images, commentaires, votes), Facebook (photos, posts, likes, suggestions d'amis) et tant d'autres.

Le premier problème d'une base de données répartie est le stockage des données. Il faut d'abord pouvoir répartir les données sur plusieurs machines car il y en a trop pour un seul serveur. Nous allons voir deux approches : l'approche de la base de données relationnelle et l'approche d'un système de fichiers distribué comme le Hadoop File System (HDFS).

Le deuxième défi est de pouvoir faire des requêtes sur les données. Les requêtes permettent de consulter les données sous une certaine forme ou alors de les transformer. Dans le cas d'un système relationnel, on utilise un langage de requêtes déclaratif comme le SQL. Dans le cas d'un système comme celui proposé dans la Lambda Architecture, on utilise plutôt MapReduce, un pipeline d'agrégation (MongoDB) ou une autre technologie pouvant être exécutée sur le cluster comme Apache Spark.

Il faut également apprendre à traiter des données en temps réel car la faiblesse des technologies de batch computing (Hadoop MapReduce, Spark) est leur temps de réponse. Parfois il faut plusieurs heures pour construire des vues à partir des données. Certains sites webs ou applications ne peuvent pas se permettre d'afficher leurs nouvelles données quelques heures en retard. La solution de la Lambda Architecture est donc d'avoir une couche d'application qui ne s'occupe que des données recentes. Cette couche utilise des technologies et base de données différentes. On parle ici de bases de données incrémentales comme MySQL, Cassandra, de système de Stream Processing comme Apache Storm et Apache Spark et d'une file distribuée comme Apache Kafka.

Bien que de nombreux concepts soient vus dans le cours, le cours a comme principal objectif de donner à l'étudiant un bon niveau en programmation distribuée. À cette fin, des devoirs d'une difficulté adéquate sont proposés à l'étudiant. Voici comment le cours va se dérouler :

En premier, on va regarder comment on fonctionne avec un système relationnel traditionnel lorsqu'on est confronté à une demande de plus en plus grande (Le Big Data). Quels sont les défis? Quels sont les problèms rencontrés? À cette fin, on va regarder la Lambda Architecture, une architecture à trois couches qui permet d'aller répondre à tous les besoins d'un système Big Data.

À l'intérieur de la Lambda Architecture, nous allons voir les technologies de bases de données ainsi que les technologies de programmation sur le Big Data (MapReduce avec MongoDB, RDD et GraphX avec Spark)

Deux langages de programmation seront utilisés : Javascript pour le MapReduce avec MongoDB et Scala pour toute la programmation sur la plateforme Apache Spark.

Certains IDE seront utilisés : IntelliJ IDE pour utiliser Spark avec le langage Scala et WebStorm pour utiliser Javascript avec Node.js et MongoDB.

Objectifs du cours

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

- Retour sur la BD relationnelle centralisée.
- Normalisation (1FN, 2FN, 3FN)
- Définir les concepts importants relatifs aux bases de données réparties
- Lambda Architecture : Batch Layer, Serving Layer et Speed Layer
- Transformation des données avec MapReduce
- Utilisation de MongoDB pour utiliser MapReduce via Javascript
- Programmation avec Apache Spark via Scala
- Programmation avec le module GraphX de Spark
- Programmation d'un Crawler avec la technologie de son choix
- Techniques de parsing pour le crawling, expressions régulières.
- Connaissance du langage Scala
- Connaissance du langage Javascript

Style d'enseignement

Le cours est donné avec une approche mixte de cours magistraux et cours pratiques. Les cours pratiques donnent aux étudiants une interaction directe avec le professeur. Cette relation directe est mieux adaptée aux défis d'apprentissage des technologies qui existent en Distributed Computing car souvent, des petits pépins mineurs liés à la difficulté de ces technologies peuvent ralentir considérablement l'étudiant. Les cours pratiques sont utilisés pour l'aide à l'installation des technologies sur la plateforme préférée de l'étudiant (Windows, Mac ou Linux). Les cours pratiques sont également populaires car les étudiants peuvent travailler régulièrement pendant toute la session sur les devoirs. Il est conseillé d'amener son ordinateur portable pour travailler en classe lors des cours pratiques (à moins de faire du pair programming).

Évaluation

L'évaluation consistera en les éléments suivants:

- Deux travaux à faire (1 à 4 personnes) à remettre les **23 octobre 2018** et **11 décembre 2018**. 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.
- Un examen intra à la séance du **23 octobre 2018**. Cet examen compte pour 25 points.
- Le devoir 1 compte pour 20 points. Remise le **23 octobre 2018**.
- Un examen final à la séance du **11 décembre 2018**. Cet examen compte pour 25 points.
- Le devoir 2 compte pour 30 points. Remise le **11 décembre 2018**.

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 18 décembre 2018.

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 6 novembre 2018.

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.

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

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.

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