

UNIVERSITÉ DE SHERBROOKE
DÉPARTEMENT D'INFORMATIQUE

IFT 630

Processus concurrents et parallélisme

Projet de session

Description du projet à remettre au plus tard le 22 juin 2018.

Projet à remettre au plus tard le 10 août 2018.

La qualité du français écrit est de rigueur, autant pour la version théorique que pratique du projet.

La charge de travail convient à une équipe de deux personnes. Annoncez cette composition aussi tôt que possible.

1 Description du projet

Dans le cadre du cours vous devez réaliser un projet qui s'aborde de façon théorique ou pratique. Une brève description de celui-ci devra être soumise pour approbation avant la date limite prescrite.

1.1 Projet théorique

Si vous choisissez un projet théorique, vous devez choisir un sujet précis, lire la documentation sur le sujet (articles, manuels, ...) et produire un rapport concis présentant le résultat de vos recherches.

La correction sera basée sur :

- Le traitement du sujet

Le sujet abordé pour le travail mérite une étude en profondeur et sous presque tous ses aspects. Votre maîtrise du sujet doit transparaître. Ne présentez pas une simple traduction des documents lus. La présentation doit être claire, concise et précise.
- Le rapport décomposé en six sections :
 - La page titre avec l'intitulé du cours et l'identification de tous les membres de l'équipe.
 - La table des matières.
 - Une introduction.
 - La revue de littérature.
 - Vos conclusions.
 - Vos références.

1.2 Projet pratique

Si vous choisissez un sujet pratique, vous devez implanter une application qui démontre l'utilisation du parallélisme.

Vous devez implanter la solution choisie, la documenter et produire un court rapport la décrivant en quelques pages.

Contraintes :

- Les programmes conçus avec les outils offerts par les laboratoires du département.
- Les programmes doivent s'exécuter sans nécessiter de paramétrisation à la ligne de commande.
- La livraison s'effectue au référentiel Git associé à votre équipe pour le projet.

La correction sera basée sur :

- Une présentation de votre application
Étant donné la variété des technologies utilisées, vous devez démontrer le bon fonctionnement de votre application.
- La qualité du code
Vous devez me remettre le code. Celui-ci doit être bien documenté, pour accélérer et augmenter sa compréhension.
- Un rapport décomposé en quatre sections :
 - La page titre avec l'intitulé du cours et tous les membres de l'équipe.
 - Le compte-rendu de votre démarche et de vos recherches.
 - La description du mode de déploiement (procédure de compilation, lancement des programmes).
 - Vos références.

2 Exemples de projets

Voici quelques exemples de projets réalisés dans le cadre de ce cours. Une description plus précise de ces projets demeure accessible sur le site web du cours.

2.1 Projet théorique

- Étude et description de Hadoop.
- Étude sur les outils de mise au point pour les programmes parallèles.
- Étude de systèmes répartis (Torque, infonuagique).
- Étude sur les Bitcoin.
- Présentation des outils disponibles pour la mise au point d'un programme *multithread*.
- Erlang : un langage de programmation parallèle.
- Étude du système de fichiers distribués GoogleFS.

2.2 Projet pratique

- Parallélisation d'un outil d'analyse génomique.
- Simulateur de livreur de pizza.
- Zsnake : une adaptation multijoueurs du jeu de Snake.
- Parallélisation de la génération d'environnements dans une application en temps réel.
- Système client-serveur de communication vocale.
- Migration d'un processus sous Windows.
- Apprentissage du blackjack par un joueur génétique.
- Système de partage de fichier entre des pairs (proche de drop box).
- Conception d'un serveur WEB en C++.
- Parallélisation du problème des N-reines.
- Parallélisation d'une portion d'un jeu de simulation de combat.
- Application d'une méthode pour démontrer la conformité d'une parallélisation.
- Algorithmes de dames parallélisés.
- Connect5.
- Parallélisation d'un algorithme à l'aide de OpenMp et Boost.
- Parallélisation d'un engin graphique.
- Génération de terrain aléatoire à l'aide de calcul parallèle MPI.
- Solutionneur de mots cachés.
- Décrypteur parallèle de mots de passe MD5.
- Le simulateur de restaurant.
- Projet SDF : Système de diffusion de fichiers.
- Implantation de méthodes asynchrones dans le langage de programmation Ruby.
- Web radio.
- Système de partage de fichiers.
- Solveur de sudoku en parallèle.
- Simulation d'une ville en construction.
- Serveur de jeu d'échecs.
- Calcul de fractales.
- Le sous-sol des Wumpus et le Monde du Wumpus.
- 100% hockey.
- Algorithme de hachage distribué.
- Crawler musical distribué.
- Simulation d'une colonie de fourmis.
- PyChat : un logiciel de clavardage.
- Plateforme de tournoi.
- Résolution de Sudoku en parallèle.
- Serveur de sémaphores distribuées.
- Briseur de mot de passe.
- Mario génétique.
- Moteur de collisions en parallèle.
- Simulation d'un département.

- Password Cracker.
- Système qui simule l'évolution boursière.
- Analyse des outils de synchronisation de Java, C#, Ruby et C++.
- Étude de l'environnement PlanetLab.
- Firebolt : un logiciel de distribution de fichier multimédia en continu.
- Analyse des applications de type Web Crawler.
- Évaluation d'applications parallèles sur des systèmes multicœurs.
- Conception d'un système distribué d'exécution.
- Un système d'exécution distribué basé sur MapReduce.
- Une solution parallèle au problème des 8 dames.
- Le grid computing et ses différents logiciels.
- Traitement parallèle pour la recherche d'images par le contenu.
- Simulateur de trafic automobile.
- Jeux en ligne massivement multijoueurs.
- Un algorithme de compression en parallèle.
- Simulation d'un hôpital.
- Roi des boids.
- Jeu d'arène multijoueur.
- Système P2P.
- Mineur de bitcoin sur Cuda.
- Simulateur de corps rigide.
- FUB Messenger, un système de chat graphique multithread.
- Solveur de fluide.