

UNIVERSITÉ DE SHERBROOKE
DÉPARTEMENT D'INFORMATIQUE

Projet de session

IFT 630

Processus concurrents et parallélisme

Description du projet à remettre au plus tard le 18 février 2023 (10%)

Projet à remettre au plus tard le 15 avril 2023 (90%)

1 Description du projet

Dans le cadre du cours vous devez réaliser un projet. Celui-ci peut être théorique ou pratique. Une brève description du projet 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.

Ce rapport doit comprendre une introduction, plusieurs sections/chapitres décrivant les éléments à présenter et une conclusion. Il devra avoir une table des matières et une bibliographie.

La correction sera basée sur :

— Le traitement du sujet

Le sujet abordé dans le travail devra l'être en profondeur et sous presque tous ses aspects. Il ne faut pas le traiter trop superficiellement. Le sujet devra aussi être traité de telle façon à ce que vous montriez une certaine maîtrise du domaine. Il ne faut pas juste traduire les documents lus. La présentation doit être claire, concise et précise.

— La qualité du français et du rapport

La qualité du français et du rapport est aussi un élément important. Le rapport doit avoir une page de présentation, une table des matières, une introduction, le corps de la présentation, une conclusion et une bibliographie. Des annexes peuvent s'ajouter pour détailler certains points du document.

1.2 Projet pratique

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

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

La correction sera basée sur :

- Une présentation de votre application
Étant donné la variété des technologies utilisées, il est nécessaire de faire une démonstration du bon fonctionnement de votre application.
- La qualité du code
Vous devez me remettre le code. Celui-ci doit être bien documenté afin d'être facilement et rapidement compréhensible.
- Le rapport
Le rapport doit écrire de façon claire et précise votre application. Il doit contenir une introduction qui présente le contexte de votre travail et votre application. Le reste du rapport décrit les technologies utilisées, le logiciel (technique, utilisation, installation), les tests effectués et les résultats obtenus. Vous devez aussi mentionner comment le parallélisme a été utilisé dans votre implantation et en quoi il a permis de produire une meilleure solution (si c'est le cas). Finalement, le rapport contient une conclusion qui effectue un retour sur le travail, indique si les objectifs sont atteints et donnent des perspectives sur des améliorations possibles.
La qualité du français sera aussi considéré dans ce rapport.

2 Exemples de projets

Voici quelque exemples de projets réalisés dans le cadre de ce cours. Une description plus précise de ces projets est disponible sur le site web du cours.

2.1 Projets théoriques

- 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 débbugger un programme multi-fils
- Erlang : un langage de programmation parallèle
- Étude du système de fichiers distribués GoogleFS

2.2 Projets pratiques

- Parallélisation d'un outil d'analyse génomique
- Simulateur de livreur de pizza
- Zsnake : une adaptation multi-joueurs 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 montrer qu'une parallélisation est correcte
- 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 fichier
- Solveur de sudoku en parallèle
- Simulation d'une ville en construction
- Serveur de jeu d'échec
- Calcul de fractales
- Le sous-sol des Wumpus et le Monde du Wumpus
- 100% Hockey
- Un briseur de hash 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 mots de passes
- 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 diffusion de fichiers multimédia en continu
- Analyse des applications de type Web Crawler
- Évaluation d'applications parallèles sur des systèmes multi-coeurs
- 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 multi joueurs
- 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 multi-fils
- Solveur de fluide
- Simulation d'un système solaire
- Calcul d'hyper-paramètres
- Comparaison de trois implantations d'un algorithme générique (parallèles vs séquentiel)
- Recherche d'itinéraires en parallèle