




5.00 crédits	30.0 h + 15.0 h	Q2
--------------	-----------------	----

Enseignants	Riviere Etienne ;
Langue d'enseignement	Anglais > Facilités pour suivre le cours en français
Lieu du cours	Louvain-la-Neuve
Préalables	Requis#: principes des systèmes informatiques, tels que visés par le cours LINFO1252
Thèmes abordés	<p>Pendant longtemps, les CPU à usage général se sont concentrés sur la prise en charge efficace d'un seul thread d'exécution. Des améliorations dans la fabrication des puces ont permis d'empaqueter plus de transistors sur la même surface d'une plaquette de silicium, et de faire fonctionner des processeurs résultants à des fréquences plus élevées. Les applications monothread s'exécuteraient simplement plus rapidement avec chaque nouvelle génération de processeur. Cette ère est maintenant terminée. L'industrie a connu plusieurs limitations connues sous le nom de mur de puissance, de mur de mémoire et de mur d'ILP. Les fabricants de puces ne sont plus en mesure d'augmenter verticalement les processeurs prenant en charge un seul thread d'exécution, ils ont commencé à regrouper plusieurs unités d'exécution plus simples, ou coeurs. L'exploitation de la puissance de plusieurs coeurs nécessite l'exploitation du parallélisme dans les applications utilisant plusieurs threads. L'écriture de code concurrent nécessite l'identification et la gestion de la concurrence, et l'introduction de la synchronisation nécessaire pour l'exactitude. L'écriture de code concurrent évolutif et performant nécessite de comprendre ce compromis entre la synchronisation et le parallélisme et de maîtriser des implémentations efficaces de structures de données partagées et d'algorithmes pour une exécution simultanée. Enfin, les processeurs multicoeurs emploient une disposition de mémoire complexe, et l'hypothèse de temps d'accès mémoire uniforme n'est plus valide. Comprendre l'impact des accès mémoire non uniformes (NUMA) est donc important pour écrire du code efficace pour les processeurs multicoeurs.</p> <p>Ce cours fournira aux étudiants les outils et les connaissances nécessaires pour écrire un code efficace et évolutif pour les processeurs multicoeurs modernes. Il détaillera les mécanismes disponibles pour la synchronisation, depuis l'implémentation de constructions de langage telles que verrous, moniteurs ou variables de condition, aux utilisations directes des primitives de synchronisation fournies par le CPU (par exemple compare-and-swap) pour construire des structures de données efficaces et évolutives. Il mettra l'accent sur les aspects de performance de la programmation multicoeur: l'impact des primitives de synchronisation, l'impact de l'accès à la mémoire non uniforme et l'impact des hiérarchies de mémoire à plusieurs niveaux. Il offrira enfin une ouverture vers l'avenir de la programmation multicoeur avec une introduction à la mémoire transactionnelle et au support de l'exécution spéculative dans les CPU modernes (par exemple Intel Haswell), et discutera du modèle d'exécution du code concurrent s'exécutant sur les GPU.</p>
Acquis d'apprentissage	<p>A la fin de cette unité d'enseignement, l'étudiant est capable de :</p> <p>Eu égard au référentiel AA du programme « Master ingénieur civil en informatique », ce cours contribue au développement, à l'acquisition et à l'évaluation des acquis d'apprentissage suivants :</p> <ul style="list-style-type: none"> • INFO1.1 • INFO3.1 • INFO6.4 <p>Eu égard au référentiel AA du programme « Master [120] en sciences informatiques », ce cours contribue au développement, à l'acquisition et à l'évaluation des acquis d'apprentissage suivants :</p> <ul style="list-style-type: none"> • SINF1.M3 • SINF3.1 • SINF6.4 <p>Les élèves acquièrent des connaissances théoriques de pointe et des compétences techniques sur les sujets abordés dans le cours.</p>

Modes d'évaluation des acquis des étudiants	Modes d'évaluation : <ul style="list-style-type: none"> • Projets et contrôle continu (40% de la note finale) • Examen sur table comportant deux parties (60% de la note finale) <ul style="list-style-type: none"> • Une première de questions de cours (questions ouvertes et QCM) • Une seconde partie portant sur un ou plusieurs articles de recherche du domaine préalablement fournis aux étudiant-es. • La proportion entre les deux parties est variable mais aucune ne dépassera 70% de la note. Les QCM n'entraînent pas de points négatifs pour une question individuelle mais la section QCM dans son ensemble peut demander un seuil minimal de réponses correctes avant de valoir des points. <p>Pour la seconde session (Aout), le projet et l'évaluation continu peuvent être refait mais uniquement de façon individuelle.</p> <p>L'évaluation des travaux soumis (projets, travaux pratiques, travail individuel, et de manière générale toute activité donnant lieu à évaluation) est basé sur la matériel (code, rapports, et tout autre matériel) soumis par l'étudiant-e, accompagné sur demande du profeseur par un examen oral complémentaire.</p> <p>Sauf indication explicite tout travail demandé doit rester confidentiel et personnel. Le partage d'information et/ou de code avec d'autres étudiants (hors binôme le cas échéant) constitue un cas de plagiat.</p> <p><u>Règles concernant l'utilisation de l'intelligence artificielle (IA) pour les activités de contrôle continu.</u></p> <p>L'utilisation de l'IA pour la réalisation des travaux pratiques et du mini-projet n'est pas encouragée car leur réalisation de façon manuelle est la meilleure manière d'acquérir les compétences visées par le cours. Si l'usage est autorisé (que ce soit pour générer du code ou pour de la documentation) il est soumis aux règles suivantes :</p> <ul style="list-style-type: none"> • Les étudiants doivent être pleinement responsables de leur production et capables d'expliquer à l'oral l'ensemble du code et des livrables (documentation, scripts de déploiement, etc.) rendus dans le cadre des travaux pratiques. • L'utilisation de l'IA doit être documentée de façon précise dans la documentation de chaque travail rendu, sous la forme d'une section dédiée indiquant quelles IAs ont été utilisées et pour quelle partie. La soumission d'un code ou d'une partie de documentation généré partiellement ou entièrement par IA mais pour laquelle l'usage n'aura pas été documenté sera considéré comme un plagiat. Les étudiants qui n'ont pas utilisé l'IA doivent aussi l'indiquer dans cette section. • Une utilisation considérée comme abusive de l'IA et mettant à mal l'acquisition des connaissances visées par le projet pourra être considérée comme une irrégularité au titre titre de la Section 7, articles 107 et suivants du Règlement général des études et des examens (RGEE), avec toutes les conséquences qui en découlent, ainsi que prévu aux articles 111 et suivants du RGEE. Lors d'une suspicion d'utilisation abusive de l'IA dans le projet rendu ou d'un rapport incomplet ou inexact de l'utilisation de l'IA, le titulaire du cours peut convoquer l'étudiant-e concerné-e à une consultation orale complémentaire, et prendre les dispositions nécessaires en accord avec le responsable des jurys EPL.
Méthodes d'enseignement	<ul style="list-style-type: none"> • Cours magistraux • Lectures et / ou production de courtes vidéos • Séances pratiques • Mini-projet
Contenu	<p>Ce cours aborde la problématique de la programmation passant à l'échelle pour les architecture multi-processeurs et multi-cœurs modernes.</p> <p>Dans une courte première partie, le cours aborde les éléments de théorie et les modèles de cohérence, et permet la compréhension des enjeux liés à la programmation concurrente en environnement en mémoire partagée. L'emphase est mise sur les aspects liés à la performance et au passage à l'échelle (utilisation de nombreux cœurs).</p> <p>Le reste du cours passe en revue un ensemble de techniques algorithmiques pour la construction de structures de données concurrentes en mémoire partagée passant à l'échelle, et étudie leurs implications en termes de performance sur les architectures modernes, en prenant en compte la hiérarchie mémoire, les protocoles de cohérence, ou les accès mémoire non uniformes (NUMA).</p> <p>Le cours est accompagné d'une série de travaux pratiques sur une machine multi-cœurs, au cours desquels les étudiant-e-s évaluent la performance et le coût de divers algorithmes et structures de données concurrents.</p>
Bibliographie	<p>The Art of Multiprocessor Programming, Maurice Herlihy and Nir Shavit, Morgan Kaufmann. ISBN 978-0-12-370591-4.</p> <p>UCL library reference 10.620.426</p>
Autres infos	<p>Tout le matériel de cours pertinent et les diapositives, ainsi que des informations pratiques relatives au cours seront accessibles sur Moodle, qui sera également le principal moyen de communication entre l'enseignant (s) et les étudiants.</p>
Faculté ou entité en charge:	INFO

Programmes / formations proposant cette unité d'enseignement (UE)				
Intitulé du programme	Sigle	Crédits	Prérequis	Acquis d'apprentissage
Master [120] : ingénieur civil en informatique	INFO2M	5		
Master [120] en sciences informatiques	SINF2M	5		
Master [120] : ingénieur civil en science des données	DATE2M	5		
Master [120] en science des données, orientation technologies de l'information	DATI2M	5		