

GROUPE	40 TREMBLAY, Guy	tremblay.guy@uqam.ca	(514) 987-3000 8213	PK-4435
	Jeudi, de 13h00 à 16h00			

DESCRIPTION	Modèles d'architectures à haute performance. Paradigmes de programmation parallèle et stratégies de conception de programmes parallèles. Métriques de performances et principales sources des surcoûts. Langages et bibliothèques de programmation parallèle. Problèmes typiques en programmation scientifique haute performance: calculs de grilles, de particules, de matrices.
-------------	---

OBJECTIFS	<ul style="list-style-type: none"> <li>• Initier les étudiant-e-s à la programmation parallèle et à certaines problématique récentes du domaine.</li> <li>• À la fin du cours, l'étudiant-e devrait être capable: <ul style="list-style-type: none"> <li><input type="checkbox"/> d'expliquer et d'utiliser les principales stratégies de programmation parallèle.</li> <li><input type="checkbox"/> de lire des programmes parallèles écrits dans quelques langages de programmation parallèle (MPD, Java, C, OpenMP, MPI) ;</li> <li><input type="checkbox"/> d'écrire des programmes dans (au moins) un langage parallèle de haut niveau (MPD, MPI ou OpenMP) ;</li> <li><input type="checkbox"/> d'écrire des programmes simples fonctionnant sur une véritable machine parallèle (MPI ou OpenMP).</li> </ul> </li> </ul>
-----------	---

ÉVALUATION	<b>Description sommaire</b>	<b>Date</b>	<b>Pondération</b>
	Deux (2) devoirs de programmation parallèle		40%
	Présentation orale sur un sujet au choix (approuvé par le professeur) et résumés écrits		30%
	Examen final		30%

L'utilisation de documentation personnelle est permise à l'examen.

Une moyenne d'au moins 50% à l'examen est exigée pour réussir le cours.

Les devoirs peuvent être réalisés seul ou en équipe de deux (2) personnes.

Une pénalité de 10 % par jour de retard sera appliquée pour la remise des devoirs.

La qualité du français sera prise en considération (jusqu'à 10 % de pénalité).

#### Politique d'absence aux examens

Un étudiant absent à un examen se verra normalement attribuer la note zéro pour cet examen. Cependant, si l'étudiant était dans l'impossibilité de se présenter à l'examen pour un motif valable, certains arrangements pourront être pris avec son enseignant. Pour ce faire, l'étudiant devra présenter à son enseignant l'un des formulaires prévus à cet effet accompagné des pièces justificatives appropriées (par ex., attestation d'un médecin que l'étudiant était dans l'impossibilité de se présenter à l'examen pour des raisons de santé, lettre de la Cour en cas de participation à un jury).

Une absence pour cause de conflit d'horaires d'examen n'est pas considérée comme un motif valable d'absence, à moins d'entente préalable avec la direction du programme et l'enseignant durant la période d'annulation des inscriptions avec remboursement : tel qu'indiqué dans le guide d'inscription des étudiants, il est de la responsabilité d'un étudiant de ne s'inscrire qu'à des cours qui ne sont pas en conflit d'horaire.

Pour plus de détails sur la politique d'absence aux examens du Département d'informatique et pour obtenir les formulaires appropriés, consultez le site web suivant :

<http://www.info.uqam.ca/enseignement/politiques/absence-examen>

CONTENU	<ol style="list-style-type: none"> <li>1. Introduction : les différentes formes de programmation concurrente (multi-contextes, distribuée, parallèle) ; les principaux paradigmes de programmation parallèle (parallélisme implicite et explicite; communication par variables partagées et par échanges de messages ; parallélisme de données, de contrôle, de flux).</li> <li>2. Aperçu des architectures parallèles : classification de Flynn ; multi-processeurs et multi-ordinateurs ; mémoire partagée et mémoire distribuée ; architectures multi-contextes.</li> <li>3. Programmation parallèle impérative : <ul style="list-style-type: none"> <li>• Notions de base : processus, tâche, <i>thread</i>, synchronisation et communication ;</li> <li>• Programmation par variables partagées : verrous, sémaphores et barrières, moniteurs ;</li> <li>• Programmation par échanges de messages : communication asynchrone ; RPC et rendez-vous ;</li> <li>• Stratégies de conception d'algorithmes parallèles : parallélisme itératif, parallélisme récursif, filtres et pipelines, parallélisme de données, sac de tâches ;</li> </ul> </li> </ol>
---------	---

- Exemples de langages : MPD, C (Threads Posix), Java, MPI, OpenMP.
- 4. Mesures de performance : temps d'exécution, coût, travail, accélération et efficacité;
- 5. Programmation parallèle scientifique haute performance : calculs de grilles et automates cellulaires ; mouvements de particules ; calculs matriciels.

## RÉFÉRENCES

- V O G.R. Andrews – *Foundations of Multithreaded, Parallel, and Distributed Programming* – Addison-Wesley, 2000. Commandé à la COOP-UQAM (Sciences).
- U O <http://www.info2.uqam.ca/~tremblay/INF7235/>  
Divers éléments (notes de cours, transparents, énoncés de devoirs) seront mis à la disposition des étudiants sur ce site.
- V C [But97] D. R. Butenhof – *Programming with POSIX Threads*. Addison-Wesley – 1997.
- V C [CG90] N. Carriero and D. Gelernter – *How to Write Parallel Programs - A First Course* – The MIT Press, 1990.
- V C [CDK+01] R. Chandra, L. Dagum, D. Kohr, D. Maydan, J. McDonald, and R. Menon – *Parallel Programming in OpenMP* – Morgan Kaufmann Publishers, 2001.
- V C [CS99] D.E. Culler and J.P. Singh – *Parallel Computer Architecture - A Hardware/Software Interface* – Morgan Kaufmann Publishers, 1999.
- V C [CT93] M. Cosnard and D. Trystram – *Algorithmes et architectures parallèles* – InterEditions, 1993. [QA76.58C68].
- A C [DG04] J. Dean and S. Ghemawat – *MapReduce: Simplified data processing on large clusters* – In OSDI'04: Sixth Symposium on Operating System Design and Implementation, pages 137–149, San Francisco, CA, Dec. 2004.
- V C [Fos95] I. Foster – *Designing and Building Parallel Programs* – Addison-Wesley, 1995. – <http://www-unix.mcs.anl.gov/dbpp>
- V C [GUD96] M. Gengler, S. Ubéda, and F. Desprez – *Initiation au parallélisme -- Concepts, architectures et algorithmes* – Masson, 1996. [QA76.58G45].
- V C [GGKK03] A. Grama, A. Gupta, G. Karypis, and V. Kumar – *Introduction to Parallel Computing (Second Edition)* – Addison-Wesley, 2003.
- V C [Han95] P.B. Hansen – *Studies in Computational Science - Parallel Programming Paradigms* – Prentice-Hall, Englewood Cliffs, NJ, 1995.
- V C [Hea02] M. T. Heath – *Scientific Computing—An Introductory Survey (2nd Edition)* – McGraw-Hill, 2002.
- V C [KK03] G. E. Karniadakis and R. M. Kirby II – *Parallel Scientific Computing in C++ and MPI* – Cambridge University Press, 2003.
- V C [KGGK03] V. Kumar, A. Grama, A. Gupta, and G. Karypis – *Introduction to Parallel Computing (Second Edition)* – Addison-Wesley, 2003.
- V C [Lea00] D. Lea – *Concurrent Programming in Java - Design Principles and Patterns (Second Edition)* – Addison-Wesley, 2000.
- V C [MSM05] T.G. Mattson, B.A. Sanders, and B.L. Massingill – *Patterns for Parallel Programming* – Addison-Wesley, 2005.
- V C [Pac97] P.S. Pacheco – *Parallel Programming with MPI* – Morgan Kaufman Publ., 1997.
- V C [Qui03] M.J. Quinn – *Parallel Programming In C With MPI And OpenMP* – McGraw-Hill, 2003.
- A C [ST98] D. B. Skillicorn and D. Talia – *Models and languages for parallel computation* – ACM Computing Surveys, 30(2):123–169, 1998.
- U C [WA99] B. Wilkinson and M. Allen – *Parallel Programming - Techniques and Applications Using Networked Workstations and Parallel Computers* – Prentice-Hall, 1999.
- U C <http://www.info2.uqam.ca/~tremblay/chercher-reference.cgi>  
Une bibliographie plus détaillée est disponible à l'URL suivant (sélection "Architecture et programmation parallèle").

A : article – C : comptes rendus – L : logiciel – N : notes – R : revue –  
S : standard – U : uri – V : volume

C : complémentaire – O : obligatoire – R : recommandé