

Note: Les éléments en italiques/rouge ne sont pas matière à examen

Liste détaillée des éléments vus en INF7235 (Hiver 2017)

3. Concepts de base de la concurrence et du parallélisme
 - Processus vs. threads
 - Concurrency vs. parallélisme
 - Graphes de dépendances des tâches et parallélisme

5. Patrons de programmation parallèles avec PRuby
 - * Parallélisme fork/join (pcall et future)
 - + Parallélisme récursif
 - * Approche diviser-pour-régner récursive
 - * Seuil de récursion pour limiter le parallélisme
 - + Parallélisme itératif (non récursif)
 - * à granularité fine (nombre "non borné" de threads)
 - * à granularité grossière (nombre limité de threads)
 - + Répartition statique par tranches d'éléments adjacents
 - + Répartition statique cyclique
 - * Parallélisme de boucles (peach et peach_index)
 - * Parallélisme de données (pmap et preduce)
 - * Parallélisme de style <<Coordonnateur/travailleurs>> (répartition dynamique)
 - * Parallélisme de flux (filtres et pipelines)

6. Métriques de performance pour algorithmes et programmes parallèles
 - * Temps, nombre de processeurs, coût, travail, optimalité
 - * Accélération, efficacité
 - * Lois d'Amdhal et de Gustafson-Barsis

8. Exemples illustrant l'approche PCAM
 - * Résolution numérique de l'équation de diffusion
 - * Opérations sur des polynômes

9. Programmation parallèle TBB/C++
 - * Approche orientée tâches plutôt que threads
 - * Notion de blocked_range
 - * Parallélisme de données: parallel_for, parallel_reduce
 - * Parallélisme de contrôle: parallel_invoke et task_group
 - * Fonctionnement du parallel_for: blocked_range, partitioner
 - * Ordonnancement des tâches et vol de tâches
 - * Approche diviser-pour-régner et notion de Range

10. Programmation parallèle avec OpenMP/C
 - * Régions parallèles et activation d'équipes de threads
 - * Distribution du travail entre threads (*sections*, boucles for)
 - * Section critique vs. opération atomique vs. réduction
 - * Création dynamique de tâches

11. *Programmation parallèle de GPU avec OpenCL/C*

- * *Architecture des GPU*
- * *Kernel vs. programme principal*
- * *Exécution SIMD*

13. Processus et échange de messages

- * Processus vs. canaux et style go
- * Calcul du min-max (trois façons)

14. Programmation MPI/C

- * Communications point-à-point vs. opérations collectives
- * Distribution des données
- * Communications asynchrones
- * Modularité et communicateurs

15. Exemples de programmes MPI/C

- * Intégration numérique
- * Calcul du min-max
- * Recherche de mots-clés
- * Diffusion de la chaleur

16. Approche PCAM de Foster

- * Partitionnement (granularité très fine)
- * Communication (analyse des dépendances de données)
- * Aggrégation (+ granularité, - communications)

18. Produit parallèle de matrices

- * Analyse des communications et distribution par blocs
- * Méthode de Mattson, Sanders et Massingil
- * *Méthodes de Fox et Cannon*

Exemples et labos

- * Nb. d'inversion
- * Histogramme
- * Polynomes
- * Intégration numérique
- * Sac de tâches distribué (mots-clés)
- * Produits de matrices (à la go, distribué)