

# Exposé oral et résumé écrit

## INF8541 — Paradigmes de programmation

Automne 2001

### 1 Objectifs

Le principal objectif de ces exposés est de vous faire découvrir divers aspects ou notions associés aux paradigmes de programmation parallèle que nous n'aurons pas eu le temps d'examiner ou d'approfondir en classe. Un autre objectif, puisqu'il s'agit d'un cours gradué, est de vérifier que vous pouvez, de façon autonome, comprendre, expliquer et synthétiser un article de recherche portant la programmation parallèle.

### 2 Ce que vous devez faire

L'exposé doit présenter, de façon claire et synthétique, un (1) article traitant de la programmation parallèle. (Voir section 5 pour quelques idées possibles de sujets.) Prenez note qu'il ne s'agit pas de faire une revue détaillée de la littérature sur le sujet, mais simplement de présenter un (1) article intéressant.

Note : Il faut faire approuver votre sujet au plus tard durant la semaine du 5 novembre, pour assurer qu'il n'y ait pas deux étudiant-e-s qui choisissent le même article.

### 3 Date et durée des exposés

Les exposés auront lieu durant les semaines du 27 novembre et du 4 décembre.

La durée (maximale!) des exposés devra être de 30 minutes. Prenez note que je ferai respecter ces contraintes de temps de façon assez stricte (à 5 minutes près).

### 4 Résumé écrit

Vous devrez me remettre, au plus tard le vendredi 7 décembre à 12:00, un résumé écrit de l'article abordé dans votre présentation. Ce résumé doit avoir *au plus 2 pages*. Il n'est pas nécessaire de fournir une page de présentation : identifiez simplement, au début de la première page, votre nom, l'article traité (auteur(s), titre, source), suivi de votre résumé. Utilisez un format raisonnable de mise en page (par exemple, la taille minimum des caractères devrait être de 10 points ou plus).

#### Quelques remarques sur ce que doit être un *résumé*

Un résumé, une synthèse d'un article ne consiste *pas* à extraire un certain nombre de phrases de l'article et à les traduire ou retranscrire pour en arriver à remplir *n* pages. En fait, toute phrase tirée (ou traduite) textuellement du texte original devrait toujours être *citée* de façon explicite, c'est-à-dire en indiquant clairement qu'il s'agit d'une citation (par exemple, à l'aide de guillemets et en indiquant la page).

Écrire un résumé synthèse consiste à lire un article, à le comprendre, et ensuite à l'expliquer *dans vos propres mots*, pour en décrire les éléments clés, les principales conclusions (l'esprit et non la lettre). Ainsi, il devrait être possible de produire un résumé comportant n'importe quel nombre de pages (1/2, 2, 5 pages) : il s'agit alors simplement de choisir le niveau de détail approprié, de déterminer ce qui est important et ce qui ne l'est pas étant donné les contraintes d'espace.

Notez que les mêmes remarques s'appliquent aussi quant à l'exposé, qui ne devrait pas dépasser 30 minutes (c'est 10 minutes de plus que ce qui est alloué dans de nombreuses conférences!).

## 5 Quelques pistes possibles

Voici quelques thèmes possibles avec quelques références :

- Programmation concurrente orientée objets [AWY93, AFK<sup>+</sup>93, Gri93, CGH94, CR96]
- Problèmes avec les *threads* de Java [Pa99, Jak00]
- Programmation Java haute-performance [AFT<sup>+</sup>00, YC97]
- Bibliothèques de *threads* [Aa96, LFA98]
- Parallélisme de données [QH90, Ba93, Lov93, RRH92, NPGM00, Bra00]
- Échange de messages [SGDM94, DM98]
- Langage de coordination [CG89, Ca95]
- Langage parallèle indépendant de l'architecture sous-jacente [DDO<sup>+</sup>92, Sny93, PJM96, Ca00]
- Langage parallèle basé sur C pour le parallélisme récursif [FLR98, VH99]
- Compilation pour machines parallèles [GC92]
- Prototypage pour systèmes concurrents [Has00]
- Modèles de consistance mémoire ou mémoire virtuelle partagée [AG96, PTM96, Hil98]
- Taxonomie des langages de programmation parallèle [ST98]
- Programmation fonctionnelle [Wik96, BNA91]
- Programmation logique parallèle [Cla88]

Vous pouvez évidemment choisir d'autres sujets ou articles que ceux indiqués plus haut. Pour des suggestions, venez me voir et, en fonction de vos intérêts, je tenterai de vous faire des propositions.

## 6 Site *web* pour choix du sujet et de la date de présentation

Pour choisir votre sujet ainsi que la date de votre présentation, consultez le site suivant:

<http://www.info.uqam.ca/~tremblay/INF8541/Expose/>

## 7 Site *web* pour bibliographies

Un site *web* est disponible pour consulter (par l'intermédiaire d'un script `cgi`) ma bibliographie personnelle sur la programmation et les architectures parallèles :

<http://www.info.uqam.ca/~tremblay/chercher-reference.cgi>

Vous pouvez aussi trouver de nombreuses références sur le site *web* suivant :

<http://www.cs.rit.edu/~ncs/parallel.html>

## Références

- [Aa96] C. Amza and al. TreadMarks: Shared memory computing on networks of workstations. *IEEE Computer*, 29(2):18–28, Feb. 1996.
- [AFK<sup>+</sup>93] G. Agha, S. Frolund, W.Y. Kim, R. Panwar, A. Patterson, and D. Sturman. Abstraction and modularity mechanisms for concurrent computing. *IEEE Parallel and Distr. Technology*, 1(2):3–14, May 1993. [Tiré de [ST95]].
- [AFT<sup>+</sup>00] Y. Aridor, M. Factor, A. Teperman, T. Eilam, and A. Schuster. Transparently obtaining scalability for Java applications on a cluster. *Journal of Parallel and Distributed Computing*, 60(10):1159–1193, 2000.
- [AG96] S.V. Adve and K. Gharachorloo. Shared memory consistency models: A tutorial. *IEEE Computer*, 29(12):66–76, 1996.
- [AWY93] G. Agha, P. Wegner, and A. Yonezawa. *Research Directions in Concurrent Object-Oriented Programming*. The MIT Press, 1993. [QA76.64R48].
- [Ba93] G.E. Blelloch and al. Implementation of a portable nested data-parallel language. In *4th ACM PPOPP*, pages 102–111. ACM, SIGPLAN Notices, 1993.
- [BNA91] P.S. Barth, R.S. Nikhil, and Arvind. M-structures: Extending a parallel, non-strict, functional languages with state. Technical Report CSG Memo 327, MIT, March 1991.
- [Bra00] T. Braunl. Parallaxis-III: Architecture-independent data parallel processing. *IEEE Trans. on Soft. Eng.*, 26(3):227–243, 2000.
- [Ca95] N. Carriero and al. Adaptive parallelism and Piranha. *IEEE Computer*, 28(1):40–49, 1995.
- [Ca00] B.L. Chamberlain and al. ZPL: A machine independent programming language for parallel computers. *IEEE Trans. on Soft. Eng.*, 26(3):197–211, 2000.
- [CG89] N. Carriero and D. Gelernter. How to write parallel programs — a guide to the perplexed. *ACM Computing Surveys*, 21(3):323–357, Sept. 1989. [Tiré de [ST95]].
- [CGH94] R. Chandra, A. Gupta, and J.L. Hennessy. COOL: An object-based language for parallel programming. *IEEE Computer*, 27(8):13–26, 1994.
- [Cla88] K.L. Clark. PARLOG and its applications. *IEEE Trans. on Software Eng.*, 14(12):1792–1804, Dec. 1988. [Tiré de [ST95]].
- [CR96] D. Caromel and Y. Roudier. Reactive programming in Eiffel. In *Object Based Parallel and Distributed Computing*, number LNCS-1107. Springer-Verlag, 1996.
- [DDO<sup>+</sup>92] M. Danelutto, R. Di Meglio, S. Orlando, S. Pelagatti, and M. Vanneschi. A methodology for the development and the support of massively parallel programs. *Future Generation Computer Systems*, 8:205–220, 1992. [Tiré de [ST95]].
- [DM98] L. Dagum and R. Menon. OpenMP: An industry-standard API for shared-memory programming. *IEEE Comp. Science & Eng.*, 5(1):46–55, Jan. 1998.
- [FLR98] M. Frigo, C.E. Leiserson, and K.H. Randall. The implementation of the Cilk-5 multithreaded language. In *PLDI '98*. ACM, 1998.
- [GC92] M. Gokhale and W. Carlson. An introduction to compilation issues for parallel machines. *The Journal of Supercomputing*, 6(314):283–314, 1992.
- [Gri93] A.S. Grimshaw. Easy-to-use object-oriented parallel processing with Mentat. *IEEE Computer*, 26(5):39–51, May 1993. [Tiré de [ST95]].
- [Has00] W. Hasselbring. The programming languages and systems for prototyping concurrent applications. *ACM Computing Surveys*, 32(1):43–79, 2000.

- [Hil98] M.D. Hill. Multiprocessors should support simple memory-consistency models. *IEEE Computer*, 31(8):28–34, 1998.
- [Jak00] P. Jakubik. Multiprocessor safety and Java. <http://www.primenet.com/~jakubik/mp-safe/MultiprocessorSafe.pdf>, 2000.
- [LFA98] D.K. Lowenthal, V.W. Freeh, and G.R. Andrews. Efficient support for fine-grain parallelism on shared-memory machines. *Concurrency: Practice and Experience*, 10(3):157–173, 1998.
- [Lov93] D.B Loveman. High Performance Fortran. *IEEE Parallel & Distributed Technology*, 1(1):25–42, Feb. 1993. [Tiré de [ST95]].
- [NPGM00] L.S. Nyland, J.F. Prins, A. Goldberg, and P.H. Mills. A design methodology for data-parallel applications. *IEEE Trans. on Soft. Eng*, 26(4):293–314, 2000.
- [Pa99] W. Pugh and al. The “double-checked locking is broken” declaration. <http://www.cs.umd.edu/~pugh/java/memoryModel/DoubleCheckedLocking.html>, 1999.
- [PJM96] J. Philbin, S. Jagannathan, and R. Mirani. Virtual topologies: A new concurrency abstraction for high-level parallel languages. In *Languages and Compilers for Parallel Computing*, pages 450–464. Springer-Verlag, LNCS-1033, 1996.
- [PTM96] J. Protic, M. Tomasevic, and V. Milutinovic. Distributed shared memory: Concepts and systems. *Parallel and Distributed Technology*, 4(2):63–79, 1996.
- [QH90] M.J. Quinn and P.J. Hatcher. Data-parallel programming on multicomputers. *IEEE Software*, 7(5):69–76, 1990.
- [RRH92] A. Rogers, J. Reppy, and L Hendren. Supporting SPMD execution for dynamic data structures. In *Languages and Compilers for Parallel Computing*, pages 192–207. Springer-Verlag, LNCS, 1992.
- [SGDM94] V.S. Sunderam, G.A. Geist, J. Dongarra, and R. Manchek. The PVM concurrent computing system: Evolution, experience, and trends. *Parallel Computing*, 20(4):531–545, 1994.
- [Sny93] L. Snyder. Foundations of practical parallel programming languages. In *Parallel Computation*, pages 115–134. Springer-Verlag, LNCS-734, 1993.
- [ST95] D.B. Skillicorn and D. Talia. *Programming Languages for Parallel Processing*. IEEE Computer Society Press, 1995.
- [ST98] D.B. Skillicorn and D. Talia. Models and languages for parallel computation. *ACM Computing Surveys*, 30(2):123–169, 1998.
- [VH99] V.-Y. Vee and W.-J. Hsu. Applying Cilk in provably efficient task scheduling. *The Computer Journal*, 42(8):699–712, 1999.
- [Wik96] C. Wikstrom. Implementing distributed real-time control systems in a functional language. In *IEEE Workshop on Parallel and Distributed Real-Time Systems*, pages ?–? IEEE, April 1996.
- [YC97] Weimin Yu and Alan L. Cox. Java/DSM: A platform for heterogeneous computing. *Concurrency — Practice and Experience*, 9(11):1213–1224, Dec. 1997.