Submitted to Parallel Matrix Algorithms and Applications - Marseille - 2004, 20-22 October

Tradeoff to minimize extra-computations and stopping criterion tests for parallel iterative schemes

O. Beaumont[1], E.M. Daoudi[2*], N. Maillard[3], P. Manneback[4], J.-L. Roch[5*]

[1] Olivier.Beaumont@labri.fr, [2] mdaoudi@sciences.univ-oujda.ac.ma, [3] nmaillard@inf.ufrgs.br, [4] Pierre.Manneback@fpms.ac.be, [5] Jean-Louis.Roch@imag.fr.

In general, iterative schemes consist in iterating some computation until a certain boolean condition is carried out. Unfortunately, for the parallel implementation on distributed memory systems, the stopping criterion test is often achieved by global synchronization which penalizes performances, in particular in the context of grid computing.

In order to mitigate this problem, different approaches have been proposed and studied in the literature. A first approach consists in gathering all global synchronizations of one iteration in order to perform them together; this technique is used in [1] for Conjugate Gradient method, but it does not allow to reduce the number of stopping criterion tests. A second approach, which has been widely studied, consists in desynchronizing communications and iterations, consequently resulting in an asynchronous algorithm[2, 3]. Unfortunately, in this context, the iteration scheme is no more preserved; thus, a critical problem is to ensure convergence. A third approach, called k-***steps*** in the sequel, is often used in practice; it consists in performing the stopping tests every k iterations only (i.e. stopping tests are evaluated at iterations $k, 2k, \dots, q.k, \dots$), till the stopping test is carried out. Let n_* denotes the exact number of iterations obtained by the standard algorithm which performs the stopping criterion test after each iteration. Then, the k-***steps*** method performs $\#I = n_* + k$ iterations and $\#T = \lceil n_*/k \rceil$ tests. However, since n_* is unknown a priori, the problem is to choose k in order to obtain a good trade-off between $\#I$ and $\#T$.

In this work, we propose a new control technique, original to the best of our knowledge, which delivers the result of iteration n_* after a small number $\#T = \log^{1+o(1)} n_*$ of stopping criterion tests while computing the result of $\#I = n_*(1+\epsilon)$ iterations only. The advantage of our technique lies in the fact that the structure of the initial algorithm is not modified, and consequently the convergence is ensured.

Our method is based on an amortized technique inspired from Floyd's and Brent's algorithms to detect periodicity in a sequence [4]. It consists in computing two numbers n_1 and n_2 such that $n_1 \leq n_* \leq n_2$, and then determining the exact value of n_*. Like the k-***steps***, our method performs only a limited number of stopping tests, however tests are not performed at regular steps anymore, but rather at steps $\rho^{f(0)}$, $\rho^{f(1)}$, $\cdots$, $\rho^{f(k)} = n_2$, where $1 < \rho < 2$ and f satisfies $\forall i,\ f(i) \leq i$. Determination of f is based on a tradeoff between $\#I$ and $\#T$. In this paper, we provide asymptotic results for different choices of f: namely $f(i) = i^\alpha$, $0 < \alpha \leq 1$; $f(i) = \frac{i}{\log i}$ and $f(i) = \frac{i}{\log^* i}$. We prove that n_* can be determined with very little more than $\#T = \log n_*$ tests – which is a lower bound for $\#T$ – while performing an asymptotic optimal number of iterations $\#I = n_* + o(n_*)$.

References

[1] Baserman (A.), Reichel (B.) et Scheltoff (C.) - Preconditioned CG methods for sparse matrices on massively parallel machines - Parallel Computing 23 (1997).

[2] Bertsekas (Dimitri D.) et Tsitsiklis (John N.) - Parallel and distributed computation, numerical methods - Prentice-Hall, Englewood Cliffs N.J. 1989.

[3] El Baz (Didier) - An efficient termination method for asynchronous iterative algorithms on message passing architectures - LAAS Report No , In proceedings of the International Conference on Parallel and distributed Computing Systems, Dijon, Volume 1, 1996.

[4] Knuth (Donald E.) - The Art of Computer Programming - Vol. 2 - 3rd edition - §3.1 p.7 - Addison Wesley Longman - 1997.

*This work is supported by Comité Mixte Franco-Marocain - Action Intégrée MA/01/19.