Optimal Mapping for Optimal tiling

B.SIDI-BOULENOUAR, H.BOURZOUFI, R.ANDONOV
Science computer department
Valenciennes university
LAMIH, UMR 8530, ISTV Le mont Houy BP 311 –
59304 Valenciennes Cedex
FRANCE

Abstract: Iteration space tiling is a technique used by parallelizing compilers to increase the computation to communication ratio by varying the granularity of the computation. The source problem that we investigated are modeled by orthogonal uniform recurrences defined over a three-dimensional, parallelogram shaped iteration space. This paper studies the best partitioning and good mapping of iteration space on a ring and on a grid of processor. The results are validated by exhaustive experiments on distributed memory machine (INTEL Paragon).

Key-Words : partitioning, communication/computation, ring, grid, mapping CSCC'99 Proc.pp.5411-5415

1 Introduction :

Tiling the iteration space is a common method for improving the performance of parallel loop programs executed in SPMD (Single Program Multiple Data) fashion on a DMM (distributed memory machine). It may be used as a technique in parallelizing compilers.

These techniques are developed very actively these last years around the automatic parallelization of the nest of [4,7,11]. Tiling consists in partitioning the iteration space into blocks called tile; a single processor executes each tile in an atomic way.

Optimal tiling consists in determining the optimal parameters of the tile (shape and size) which to minimize the execution time by reducing the extra cost of communications.

In the case of two-dimensional uniforms dependency loops, the results described in [1,2,6,9] propose an analytical solution with the optimal size of the tile. The objective of this work is to develop tiling techniques in the case of a three-dimensional uniform recurrence and to propose best architecture (ring or grid) to project the iteration space. This last problem according to our knowledge was never studied.

2 Tiling :

In the field of the parallel programming, partitioning holds a significant place. Indeed parallelization with fine grain can lead to very bad results because of the overcoats due to the communications. The solution is to gather elementary operations to link communications, i.e. partitioned the field of computationation in tiles. A processor will carry out each tile in an indivisible way.

One of the major objectives in this search is to find optimal parameters(shape, size) of the tile for a better computation/communications ratio. In the case of two-dimensional uniform recurrence, the problem of partitioning have be largely studied [4,7,11], what be not the case of three-dimensional.

The approach used in this paper is based on the results presented in [1,2,3]. The optimal size of the tile is obtained by minimizing the total execution time, which is a function whose dimensions of the tile are variables. The other parameters of the function are the following: the size of space of iteration, the volume of communication, the number of processor and the design features techniques of machine (the time of establishment of a communication (β), the time of computationation of a elementary data (τ_a), the time of transmission of an elementary data (τ_t).

Let us consider the following example (E1), That is to say a nest of loops of depth three; n_1, n_2 and n_3 dimensions of the space of iterations and F a function of recurrence with orthogonal dependencies : $d_1=(1,0,0)$ and $d_2=(1,1,3)$;

```
For i=1 to  $n_1$
  For j=1 to  $n_2$                              (E1)
    For k=1 to  $n_3$
      f[i,j,k]=Max{f[i-1,j,k];f[i-1,j-1,k-3]}
    end
  end
end.
```

We can find this type of dependencies in many applications resulting from the dynamic programming. One of most interesting is generalization 3D of the algorithm of Smith and Waterman for the comparison of the biological sequence[12].

There are two ways to project the iterations space (E1) either on a grid or on a ring of processors.

3 Projection on a grid :

The approach that is used is based on the results presented in [3]. We give here only the principal steps of that approach:

-Partition the iteration space into orthogonal tiles (rectangular parallelepipeds) of size (r*s*t) and leave r, s and t free variable. Each processor receive a surface of size (r*t) of its neighbor North and a surface of size (s*t) of its neighbor West, calculate its tile and send a surface of size (r*t) to neighbor South and a surface of size (s*t) to neighbor East (Fig.1).

-Cluster the recurrence into tiles, yielding a new uniform recurrence over a new domain. Each tile is considered as an atomic computation.

- Apply systolic space-time transformations, yielding a virtual two-dimensional systolic array.
- Implement this array on a torus of size (q_1*q_2 = p) processors.
- Relax the systolic space-timing model to account for practical machines, and obtain a formula for the total running time of the final implementation. This formula will be expressed as a function of r, s and t, as well as other parameters of the program and architecture.
- The optimal tile size is obtained by minimizing the total execution time:

$$T(r,s,t) = \frac{n_1 n_2 n_3}{q_1 q_2 rst}(4\hat{a} + \hat{o}_a rst) + (q_1 - 1)(2\hat{a} + \tau_t st + \hat{o}_a rst)$$

$$+ (q_2 - 1)(2\hat{a} + \hat{o}_t rt + \hat{o}_a rst) \qquad (1)$$

Under the contraintes :

$$1 \le r \le n_1 / q_1 \quad et \quad 1 \le s \le n_2 \quad 1 \le t \le n_3$$

$$\min(q_1(2\beta + \tau_t st + \tau_a rst), q_2(2\beta + \tau_t rt + \tau_a rst)) \le \frac{n_3}{t}(4\beta + \tau_a rst)$$

The optimal size of the tile is given by:
Where:

$$(r_{opt}, s_{opt}, t_{opt}) = \begin{cases} (\frac{n_1}{q_1}, \frac{n_2}{q_2}, t_{opt}) \quad if \ \lambda_1 > 0 \quad (no\,cylic\ solution) \\ r_{opt}, s_{opt}, 1) \quad else \quad (cyclic \quad solution) \end{cases} \qquad (2)$$

$$sopt = \sqrt{\frac{q2 - 1}{q1 - 1} v*}$$

and optimal volume is given by :

$$v* = \sqrt{\frac{4\beta n_1 n_2 n_3}{q_1 q_2 (q_1 + q_2 - 2)\tau_a}}$$

In a broad spectrum of problems, our objective is to evaluate the profit between the cyclic solution and the no cyclic solution in the concrete case of INTEL Paragon machine.

Comparison between the solution cyclic and no cyclic :

$$ropt = \sqrt{\frac{q1 - 1}{q2 - 1} v*}$$

The comparison between the cyclic and no cyclic version was tested on the first six problems among the seven presented in the following table(Table1) (the last problem is used for the ring version). Each problem represents an authority with various values of n_1, n_2 and n_3 in order to obtain various forms for iteration space:

Pbs	n_1	n_2	n_3
pb1	500	500	500
pb2	50000	10	50 000
pb3	500	500	40 000
pb4	50000	50000	10
pb5	30000	2000	20
pb6	50000	3000	100
pb7	500000	10	10

Table1: Data

The table(Table2) is divided into two parts: the first (column 3,4,5,6) corresponds to the no cyclic solution and the second (col 7,8,9,10) to cyclic solution.

The choice of the solution depends directly on the value of λ_1 (col 2). Indeed, if λ_1 is positive, the optimal solution is not cyclic. Otherwise the optimal solution is cyclic.

In the two parts, we calculate the theoretical and experimental solution.

For the theoretical part of the no cyclic version, we calculate t_{opt} according to the formula (). If t_{opt} is higher than one, the theoretical solution is kept; if not, as t_{opt} cannot be lower than one, we recomputed optimal time by realizable approximate t_{opt} equal to one (collar 5). In the sixth column, we give the experimental time.

$$t_{opt} = \sqrt{\frac{4\beta n_3 q_1 q_2}{\tau_a n_1 n_2 (q_1 + q_2 - 2) + (q_1(q_1 - 1)n2 + q_2(q_2 - 1)n_1)\tau_t}};$$

The second part of table (cyclic) is also divided into a theoretical part (collar 7,8,9) and an experimental part (collar 10).

In the seventh column, we calculate t_{opt}, the eighth column shows the execution time obtained by the cyclic solution. The last but one column present passes number. In the last column, we give the experimental optimal value for the cyclic solution.

We note that the theoretical value for the two solutions is close to the experimental value. We observe thus that for a square field or a field whose third dimension n_3 is larger than n_1 and n_2, the no cyclic solution has a modest advantage on the cyclic solution. In the other case, the cyclic solution is shown more interesting than the no cyclic solution. The previous results show that the non-cyclic version is not always that allows to obtain the optimal time. Contrarily to conclusions of Ohta and al[8], the preceding table shows that for certain problems the cyclic version is the best.

Conclusion

When mapping is done according to smallest dimension, it is always interesting to use the cyclic solution.

4 Projection on a Ring:

The space of iterations (E1) is partitioned in rectangular parallelepipeds of size r*s*t , Each tile is considered as an atomic computation. The mapping of the graph of tiles is carried out vertically on a ring of p processors. Each processor receives a surface s*t of its neighbor of left-hand side, calculates a tile and sends a surface of size s*t to its neighbor of right-hand side.

The execution time of a tile is $\tau_a rst$, the number of tiles is : N1*N2*N3 with N1=n1/r, N2=n2/s, N3=n3/t. While following the approach[2], the optimal size of the tile is obtained by minimizing the time execution total :

$$T(r,s,t) = \begin{cases} N_1 N_3 P_t \geq pLt \rightarrow \\ \qquad T_1(r,s,t) = \dfrac{N_1 N_2 N_3 P_t}{p} + (p-1)L_t \\ N_2 N_3 P_t \leq pL_t \rightarrow \\ \qquad T_2(r,s,t) = N_2 N_3 P_t + (N_1 - 1)L_t \end{cases} \quad (3)$$

Under Contraints:

$$1 \leq r \leq n1/p; \quad 1 \leq s \leq n2; \quad 1 \leq t \leq n3$$

The execution time (according to r and s and t) is given by the function:

$$T(r,s,t) = \frac{2n_1 n_2 n_3 \beta}{prst} + (p-1)\tau_a rst + (p-1)\tau_t st + (p-1)\beta$$

$$\frac{n_1 n_2 n_3 \tau_a}{p} \quad (4)$$

The size of the message sent (received) is constant and is equal to s*t. We put s*t=h, we replace in equation(4); the passage of a graph of tiles 3D towards a graph of tiles 2D is given by the figure (Fig.2(b)).

$$T(r,h) = \frac{2n1n2n3\beta}{prh} + (p-1)\tau arst + (p-1)\tau tst + (p-1)\beta$$

$$+ \frac{n1n2n3\tau a}{p} \quad (5)$$

The resulting equation(5) is a model for two dimensional uniforms recurrences. We can apply the results described in [2,1,5].

The optimal size of a tile is given by:

$$(h_{opt}, r_{opt}) = \begin{cases} \left[\sqrt{\dfrac{2pn_2 n_3 \beta}{(p-1)n_1 \tau_a}}, \dfrac{n_1}{p} \right] & if \quad \lambda 2 > 0 \\ & (no\ cyclic\ solution) \\ \left[1, \sqrt{\dfrac{2n_1 n_2 n_3 \beta}{(p-1)p\tau_a}} \right] & else \\ & (cyclic\ solution) \end{cases} \quad (6)$$

with $\lambda 2 = 2pn_2 n_3 \beta - (p-1)n_1 \tau a$

Then, we will are interest in the evaluation of the difference between the cyclic and no cyclic solution.

Comparison between the cyclic and no cyclic solution :

In the first part of table(Table3) (column 3,4,5,6), we present the no cyclic solution, while in the second part (col 7,8,9) we give the cyclic solution. For the seven problems (table1} presented previously, we calculate the value of λ_2. If the value of λ_2 is positive, we will adopt the no cyclic solution if not the cyclic solution.

In column3, we calculates optimal surface h_{opt} according to the formula(6). If h_{opt} is higher than one, we will adopt the theoretical solution (collar 4) if not, we recomputed the optimal solution with h_{opt} equalizes with one(col 5). In column 6, we give the experimental results of the no cyclic solution. We calculate the optimal width of the tile (r_{opt}) in column 7. We give the optimal execution time of this version in column 8. Finally, column 9, represents the experimental results of the cyclic solution. We note that the experimental results remain close to the theoretical results. We note thus that the cyclic solution remains limited compared to the no cyclic solution, we needed a field completely deformed (pb7) to find that the cyclic solution will take a small advantage compared to the no cyclic solution.

Conclusion :

The two solutions give very close results. To avoid the problems of buffer and for the simplicity of the no cyclic solution it is better to choose the latter.

5 Comparing the grid and ring mapping :

We compare mapping ring and mapping roasts in the point of their best behaviors. The results observed in the table below show a negligible difference between the version ring and the version roasts. In other words, the optimal solution of the grid is very close to that of the ring.

Pbs	Ring(sec)	Grille(sec)
pb1	3.39	3.36
pb2	640.45	640.04
pb3	256.87	256.60
pb4	640.45	656.01
pb5	31.24	31.16
pb6	384.78	384.46

Table1 : data

Strategy :
If we choose mapping on a ring, it is better to choose the no cyclic solution. When we choose mapping on a grid and if this one is done according to smallest dimension, it is interesting to choose the cyclic solution.

6 Conclusion :

The approach suggested in this paper develops the techniques of optimal tiling to find the best computation/communication ratio for three-dimensional orthogonal uniform recurrences. Two aspects of this problem will interest us. The first one consists in showing which of the two solutions (no cyclic and cyclic) is the best. We validate thus in experiments the results of [3]. We showed as the optimal solution can be given as well by the cyclic version as by the no cyclic version.

The second one consists in finding an optimal strategy using a fixed number of processors, which can be configured either out of ring or out of grid. We compared two methods of mapping of the space of iterations on a grid and a ring of processors. For uniform orthogonal dependencies, the results of two methods are very close. Therefore, we can deduce from it that the optimal solution can be given by the simplest strategy: configure the processors out of ring and use only one pass.

References:

[1] R.Andonov, H.Bourzoufi, and S.Rajopadhye. Two-Dimensional Orthogonal Tiling: from theory to practice. *In International Conference on High Performance*
Computing (HiPC), pages 255-231, 1996. India, IEEE Computer Society Press.

[2] R.Andonov and S.Rajopadhye. Optimal Orthogonal Tiling of 2-D Iterations. *Journal of parallel and distributed Computing*, 45: 159-165, September 1997

[3] R.Andonov, N.Yanev, and H.Bourzoufi. Three-dimensional Orthogonal Tile Sizing Problem: Mathematical approach. In L. Thiele, J. Fortes, K.Visser, V. Taylor, T.Noll and J.Teich, editors, International Conf. *On Application Specific Systems, Architectures and Processors ASAP'97*, pages 209-218. IEEE, 1997. Zurich, Switzerland.

[4] P.Boulet, A.Darte, T.Risset and Y.Robert. (Pen)-Ultimate Tiling? *INTEGRATION, the VLSI journal*, 17:33-51, Nov 1994.

[5] H.Bourzoufi, B. Sidi Boulenouar et R.Andonov. Pavage optimal pour des dépendances type sac à dos. *Renpar'10*, pages 161--164, 1998.

[6] S.Hiranandani, K. Kennedy, and C.Tseng. Evaluation Compiler Opimizations for Fortran D. Journal of parallel and Distributed Computing, 21 :27-45, 1994.

[7] F.Irigoin and R.Triolet. Supernode Partitioning. *15th ACM Symposium on Principales of Programming Languages*, pages 319--328. ACM, Jan 1988.

[8] H.Ohta, Y.Saito, M.Kainaga, and H.Ona. Optimal tile size adjustement in compiling general DOACROSS loop nests. *In International Conference on Super-computing*, pages 270--279, July 1995.

[9] D.Palermo, E.~Su, A.Chandy, and P.Banerjee. Communication Optimizations Used in the PARADIGM Compiler for Distributed-Memory Multicomputers. *International Conference on Parallel Processing*, St.Charles, IL, August 1994.

[10] P.Quinton and Y.Robert. *Systolic Algorithms and Architectures*. Masson, 1991

[11] J.Ramanujam and P.Sadayappan. Tiling Multidimensional Iteration Spaces for Non Shared-Memory Machines . *Supercomputing 91*, pages 111--120, 1991.

[12] M.S Waterman. Mathematical Methods for DNA Sequences. *CRC Press*, Inc. Boca Raton, Florida, 1989.

No Cyclic		Cyclic	
Theoretical	Exp	Theoritical	

Pb	λ_1	t_{opt}	$T(n_1/q_1, n_2/q_2,t_{opt})$	$T(n_1/q, n_2/q,1)$		r_{opt}	$T(r_{opt}, n_2/q_2,1)$	Nb Pass	/
Pb1	13.37	2.00	**3.36**	3.39	137		3.40	/	/
Pb2	1437.95	14.15	**640.044**	/	647.123	516	1073.53	/	/
Pb3	1150.98	17.96	**256.60**	/	267.68	410	259.534	/	/
Pb4	-10199.7	0.05	~~640.03~~	1083.76	/	516.02	**565.01**	24	664.23
Pb5	-244.23	0.02	~~31.15~~	41.31	/	241.53	**31.16**	31	32.17
Pb6	-609.13	0.03	~~384.64~~	409.313	/	554.15	**384.46**	49	389.75

Table2 : grid results (q_1=5, q_2=4)

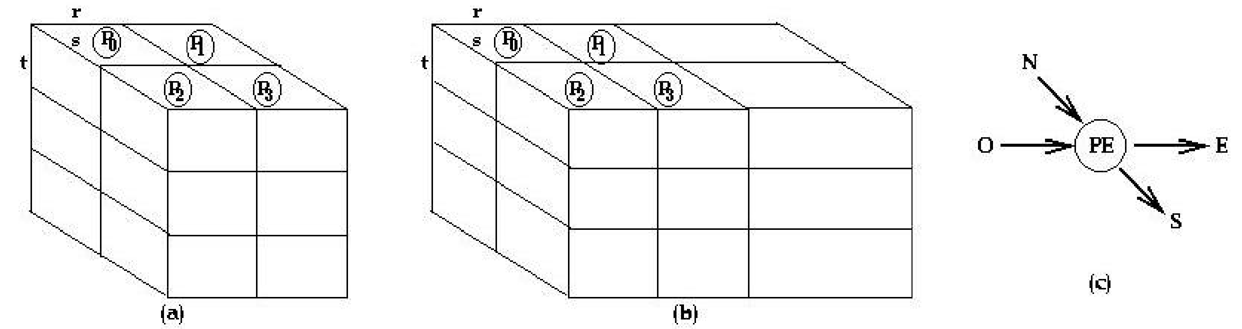

Fig.1: Grid No cyclic solution(a), cyclic solution(b), communications(c)

		No Cyclic				Cyclic		
		Theoretical			Exp(sec)	Theoretical		Exp(sec)
Pbs	λ_2	h_{opt}	$T(n_1/p,h_{opt})$ (sec)	$T(n_1/p,1)$ (sec)		r_{opt}	$T(r_{opt},1)$ (sec)	/
pb1	899.995	430.81	3.394	/	3.49	10774.9	3.397	/
pb2	1799.52	60.95	640.453	/	651.31	152380.0	640.455	/
pb3	72000.0	3853.29	256.871	/	262.98	96373.9	256.869	/
pb4	1799.52	60.9519	640.455	/	649.34	152380.0	640.455	/
pb5	143.709	22.25	31.240	/	32.20	33384.9	31.243	/
pb6	1079.52	47.21	384.781	/	393.53	118033.0	384.782	/
pb7	-4.485	0.27	1.40	1.53	/	6814.66	1.40	1.51

Table3 : Ring Results (p=20)

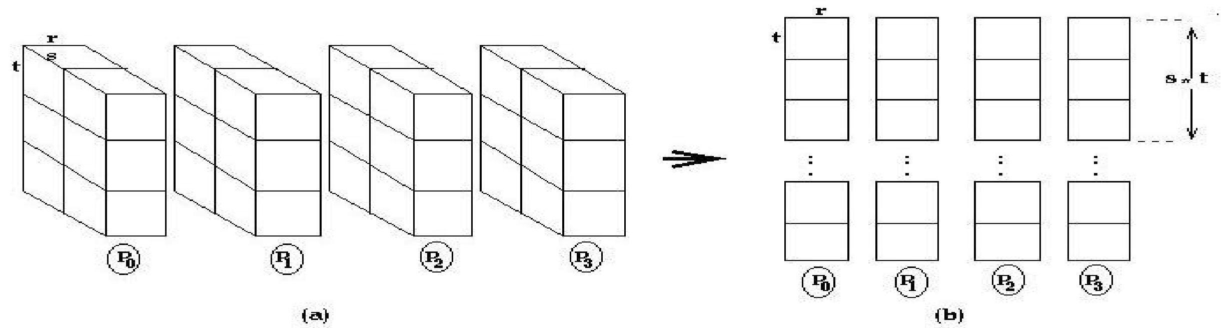

Fig2 Ring: transformation of a tile graph 3D (a) into a graph 2D(b)