PARALLEL CALCULATION OF 3-D PARASITIC RESISTANCE AND CAPACITANCE WITH LINEAR BOUNDARY ELEMENTS¹

Wenming Zhou, Zeyi Wang and Lan Rao

Dept. of Computer Science & Technology Tsinghua University Beijing 100084, China Tel: +8610-62785564 Fax: +8610-6259 e-mail:wangzy@tiger.cs.tsinghua.edu.cn

Abstract--The widespread application of deep sub-micron and multilayer routing techniques makes the interconnection parasitic influence become the main factor to limit the performance of VLSI circuits. Parallel direct boundary element calculation of three-dimensional (3-D) resistance and capacitance is an important method for fast extraction. In this paper, a parallel algorithm to implement linear element calculation by using PVM (Parallel Virtual Machine, a distributed calculating software on PC network) is introduced. The hierarchical calculation scheme of the setup and solution processes of linear equations is discussed. At the end, the performance and workload balance of the algorithm are analyzed.

1. INTRODUCTION

In VLSI circuits, because of the application of deep sub-micron and multilayer metal routing techniques, the interconnection parasitic influence has more and more effect on the performance such as circuit delay, power consumption, etc. Nowadays, design of VLSI circuits must include the correct evaluation of interconnection characters.

From the beginning of 1990s, discussion of the boundary element method (BEM) calculation of 2/3-D parasitic resistance and capacitance is very active [2,3,4,5,13,14,17]. The excellence of the BEM lies on its accuracy, calculation speed and adaptability for complex structures. K. Nabors et al. successfully [13,14] apply the multipole accelerated algorithm of integral equation to 3-D capacitance extraction, thus it makes the boundary element method be more attractive. Besides, developing the inner parallelism of the BEM is another important access to improve the calculation speed, especially for 3-D extraction with tremendous quantity of computation.

The paper [15] assigns sub-structure to parallel CPUs with sub-structure BEM. But it is very difficult to get a balanced partitioning of the sub-structure for complicated geometry shape. The literature[16] maps boundary elements to each CPU in a MIMD parallel machine. It is easy to suit a parallel system with any amount of computational units.

The paper [6] used the approach like [16] for the constant element calculation and the Gauss-Jordan elimination of linear equation set and got good parallel performance.

Based on [6], this paper presents a parallel algorithm suitable for distributed calculation supported by the PVM parallel environment (MIMD type). It uses the linear boundary elements and the iterative method GCR[11] to solve linear equation set. Section 2 introduces the calculating problem and parallel environment. Section 3 discusses the scheme for parallel implementation of two main steps in direct boundary element method. The analysis of results is given at the end.

2. PROBLEM AND COMPUTATIONAL ENVIRONMENT

The mathematical model of the VLSI parasitic resistance and capacitance is the Laplacian equation with mixed boundary condition. Applying the Green formula and characters of the fundamental solution of the Laplacian equation on it, following direct boundary integral equation [2,3,4] is gotten:

$$C(P)u(P) + \int_{\Gamma} u(Q)q^{*}(P,Q)d\Gamma_{Q} = \int_{\Gamma} q(Q)u^{*}(P,Q)d\Gamma_{Q}$$
(1)

where the boundary Γ of region Ω consists of forced boundary Γ_u and free boundary Γ_q , i.e. $\Gamma = \Gamma_u + \Gamma_q = P$ is source point on Γ , Q is field point. C(P) is a constant dependent on the neighboring geometry of P on the boundary Γ . u(P) is the potential at source point P, q(Q) and u(Q) are the normal electric field and the potential on Γ . In 3-D situation, u*(P,Q) is the fundamental solution as follows:

$$u^{*}(P,Q) = \frac{1}{4\pi r(P,Q)}$$
(2)

where r is the distance between source point P and the field point Q, i.e.,

$$r(P,Q) = \sqrt{(x_P - x_Q)^2 + (y_P - y_Q)^2 + (z_P - z_Q)^2}$$
(3)

¹ This work is supported by National Science Foundation 60376019. ASP-DAC '97 0-89791-851-7\$5.00 © 1997 IEEE

Discreting the boundary Γ and using Gauss integration, the discreted boundary integral equations with linear elements can be represented as follows:

$$Ax=B (4)$$

where column vector x contains independent unknown variables, which number equals to that of source points [1,2,4,7].

The algorithm GCR(Generalized Conjugate Residual) method can be described as follows:

/* initialization */

 $w = B \quad X = 0$

 $/\ast$ set the criterion to stop the iteration, TOL is a constant determined by user $\ast/$

$$Err = ||w||_2$$

- $Tol = TOL\ast ||w||_2$
- iter = -1

/* GCR loop, NITER is the maximum number of iteration $^{*\!/}$

while (Err > Tol && iter < NITER)
{
iter++;

$$u^{iter} = w;$$

 $Pu^{iter} = Aw;$
/* change Pu^{iter} according to Pu^m(m=0,1,...,iter-1) */
for m = 0 to iter-1
{
 $\beta = (Pu^{iter})^{T}Pu^{m}$
 $u^{iter} = u^{iter} - \beta u^{m};$
 $Pu^{iter} = Pu^{iter} - \beta Pu^{m};$
}
/* normalize the search direction */
 $Pu^{iter} = Pu^{iter} / ||Pu^{iter}||_{2};$
 $u^{iter} = u^{iter} / ||Pu^{iter}||_{2};$
/* update X and residual w */
 $\alpha = w^{T}Pu^{iter};$
 $X = X \quad \alpha u^{iter};$
 $w = w \quad \alpha Pu^{iter};$
 $Err = ||w||_{2};$
}

PVM is a software system that permits a network of heterogeneous UNIX computers to be used as a single large parallel computer. Thus large computational problem can be solved by using the aggregate power from many nodal computers. The software version used here is PVM 3.3.

The PVM collection, which is defined by user, of serial, parallel, or vector computers appears as one large distributed-memory computer. Throughout this paper the term virtual machine will be used to designate this logical distributed-memory computer and host will designate one of the member computers. The PVM provides the function to start up tasks automatically on the virtual machine and allows the tasks to communicate and synchronize with each other. A task in the PVM is defined as a unit of computation analogous to a UNIX process. Application programs written in FORTRAN77 or C can be parallelized by using message-passing commands provided by the PVM. By passing messages among nodal computers, multiple tasks of the application program can cooperate to solve problem in parallel.

Because of its agile configuration and high ratio of performance vs. price, the PVM is very suitable to implement parallel linear element calculation of parasitic resistance and capacitance. Our code was written in parallel C programming language.

According Flynn s classification, a computing system can be classified as 4 kinds: SISD, SIMD, MISD and MIMD. The PVM belongs to MIMD type. Our algorithm must be designed according to the character of the PVM system.

In designing a parallel algorithm, workload balance and communication expense must be seriously considered. To calculate 3-D resistance and capacitance with linear boundary element method, the main process is designed as follows:

- . Automatically mesh generation
- . Formation of source points [7]
- . Set-up of linear equation set
- . Solution of linear equation set
- . Calculation of resistance and capacitance.

The mesh generation, formation of source points and calculation of resistance and capacitance are not CPU timeconsuming. So they are done in serial. The other two phases are time-consuming. But their parallelism is very good, so they are implemented in parallel.

PARALLEL CALCULATION

Set-up of Linear Equation Set

From the equation (1), every row of matrices A and B are related to a source point, every item of a row is produced with several boundary elements related to this node. Suppose that the boundary Γ is discretized into M linear boundary elements, which include N calculating source points. We send information of all M boundary elements to every host and assign all the calculating source points to each nodal computer, i.e., host, in a cyclic manner via the wrap around map [16,6]. Symbol nProc represents the total number of member processors. All source points are numbered. The wrap around map of the source points means that the source points with the number the same as that of modulo-nProc the calculating source points are assigned to the same processor. This can yield a good balanced workload. Therefore, a host only handles those calculating source points assigned to it to produce the corresponding items of matrices A and B.

Every row of matrices A and B are only related to a calculating source point. During the process there is no

need to exchange data between the hosts, because the information of all boundary elements and calculating source points has been sent to every host. After the calculation is finished, the result is sent back to main control process. Because of reducing communication, the parallel efficiency of linear system set-up is very high. As the following result showing, the efficiency of linear system set-up can be up to 90%.

Solution of Linear Equation Set

From description of the GCR in section 2, product of the matrix A and vector W is the most time-consuming part. So, it is important to balance the computational workload of the product AW. In fact, every host possesses almost the same number of rows of the matrix A after setup of the linear equation set. Thus, there is no need to distribute the matrix A again and corresponding components of the product AW can be completed in each host independently. Unfortunately, those components completed in each host must be transmitted and assembled into a complete vector AW to calculate $\beta, \| Pu^{iter} \|_2$ and α , etc. It was found that the above calculation scheme can reach very good workload balance in solution of the linear set. But it can not get good parallel efficiency because of tremendous communication consumption. In spite of this, it is believed that with the extension of the matrix size and raise of communication speed, the ratio of communication will be reduced and the efficiency of parallel algorithm will be improved.

PERFORMANCE OF PARALLEL ALGORITHM

Usually, two figures, speed-up (S_p) ratio and parallel efficiency (E_p) , are used to evaluate performance of a parallel algorithm. Workload balance is also an important factor to evaluate a parallel program.

The speed-up ratio S_P of a parallel algorithm is defined as follows:

$$S_p = T_1 / T_P \tag{5}$$

where, T_1 is the CPU time in which an algorithm is implemented by one processor, and T_P by p processor. The parallel efficiency E_P of an algorithm is defined as:

$$E_p = S_p / p \tag{6}$$

where p is number of processors used in the PVM system. The parallel performance is analyzed by an example shown in Fig. 1.

Fig.1 shows a single dielectric capacitor that consists of two metal layers and substrate. The upper three parallel conductors are named M2, the lower three ones are M1. Our task is to calculate all the capacitance between metal conductors and substrate. The boundary Γ of this parasitic capacitor is discretized into 864 boundary elements with 1238 unknown variables.

Speed-up Ratio and Parallel Efficiency

In Table 1, two columns Sp1 and Sp2 are speed-up ratio of the set-up and solution of the linear system, respectively, and Ep1 and Ep2 are corresponding parallel efficiency. Sp and Ep are overall speed-up ratio and efficiency, respectively. The runtime is given according to different host number. From Table 1, it was found that the solution time of the linear system is much less than that of the set-up. On the other hand, there are several times of communication during the linear system solution. So, compared with the linear system set-up, ratio of communication in the solution is higher and its performance is not very satisfactory. At the same time, Table 1 shows that it does not make too much loss for the whole parallel efficiency. If the host number used is reasonable, the overall performance is still very satisfactory.

 TABLE 1

 Speed-up Ratio and Parallel Efficiency in Two Calculation

 Phases and Whole Process

Р	Set-up	Sp1	E _{p1}	Solution	Sp2	Ep2	Sp	Ер
	(s)			(s)				
1	3608			66				
2	1805	1.999	0.999	42	1.570	0.785	1.989	0.994
4	906	3.982	0.995	31	2.106	0.526	3.919	0.979
6	605	5.970	0.994	25	2.654	0.442	5.383	0.973
8	454	7.951	0.993	22	2.964	0.370	7.717	0.964

Workload Balancing

The workload balancing of our algorithm in two phases of computation is shown in Fig.2. Six processors are used here.





Fig. 1 An interconnection parasitic capacitance structure.



Fig. 2 The workload balance in two phases on six processors.

Fig.2 shows that the workload is well balanced. This improves the overall performance of the algorithm.

C. Performance Curve



Fig.3 The performance curve.

Clearly, the efficiency of the algorithm degrades with increase of processor number. It is not very difficult to explain this problem. When the size of the considered problem is fixed, increase of the processors' numbers means more communication cost. If the time spent on message passing dominates the whole computation time, the efficiency degrades. So, if the problem size is comparably small, it is better to use fewer processors.

CONCLUSION

In designing the parallel algorithm for linear boundary element calculation, we conclude as:

a. To implement the algorithm on PVM, we map source point to every host based on wrap around map according to the set-up and solution of discrete boundary integral equations. It is very efficient in reducing communication cost.

b. The parallel algorithm here can well use hardware resource including different host number and avoid the demerit of region partitioning in sub-structure.

c. Workload balancing during set-up and solution of the linear system is very good, and it very useful to improve parallel efficiency.

d. During system solution, because of the tremendous communication cost, its parallel performance is not very good. There is need to improve the above parallel algorithm.

REFERENCES

- Brebbia,C.A, "The boundary element method for engineers", Prntech Press,1978.
- Wang, Q.Wu, "A new approach treating corners in boundary element method", 13th International Conference on Boundary Element Methods, pp.901-911, 1991
- Wu, Z. Wang "Application of boundary element method in VLSI CAD" Computational Physics (in chinese) 1992.9
- Zeyi Wang and Qiming Wu, "A two-dimensional resistance simulator using the boundary element method", IEEE Trans. on CAD, Vol.11,No.4,pp.497-504,1992
- V.Rohklin, Rapid solution of integral equations of classical potential theory", J. of Computational Physics, Vol.60, pp.187-207, 1985
- YanhongYuan, Zeyi Wang and Qiming Wu, "Parallel boundary element calculation of three-dimensional resistance and capacitance", Proc. of CAD/Graphics 93, New Advances in Computer Aided Design and Computer Graphics, Beijing, Aug.1993
- Wenming Zhou, "Algorithm for linear element calculation of 3-D parasitic resistance and capacitance and its parallel implementation", MS Thesis of Tsinghua University 1996
- Jin-Song Hou, Zeyi Wang and Chunling Zhou, "The calculation of twodimensional resistance and capacitance with realistic shapes", Proc. of Microelectronic Package and PCB Technology, Beijing, pp108-112, Sep.19-23, 1994
- C.Patterson and M.A.Sheikh, "Interelement continuity in the boundary element method", Topics in Boundary Element Research, (Ed. Brebbia, C.A), Vol.1, pp123-141, Springer-Verlag and Berlin, 1984
- Dongxiang Wu, "Boundary element calculation of multimedia 3-D parasitic resistance and capacitance", B.S Paper of Tsinghua University 1995.
- 11. Yanhong Yuan, "Research of parallel algorithm for 3-D parasitic resistance and capacitance", M.S Paper of Tsinghua University, 1994
- Oak Ridge National Laboratory, "PVM 3 users guide and reference manul", May. 1995
- K.Nabors and J.White, FastCap: "A multipole accelerated 3-D capacitance extractor program", IEEE Trans. on CAD, Vol.10, No.11, pp.1447-1459, Nov. 1991

- K.Nabors, S.Kim and J.White, Fast capacitance extraction of general three-dimensional structure", IEEE Trans. on CAD, Vol.40, No.7, pp.1496-1506, 1992.
- 15. K.A.Kline, et al. "Parallel processing and the solution of boundary element equations".
- Alberto Zucchini et al., "Vectorial and parallel processing in stress analysis with the boundary element method", International Journal for Numerical Methods in Engineering, 31(1991), 307.
- 17. Fukude, et al., A VLSI 2-D capacitance simulator for complex structure based on actual processes", IEEE Trans. on CAD, Vol. 9, No. 1, Jan. 1990.