A Parallel Low-Rank Multilevel Matrix Compression Algorithm for Parasitic Extraction of Electrically Large Structures

Chuanyi Yang
University of Washington
Box 352500
Seattle, WA 98115
1-206-221-6513
cy@u.washington.edu

Swagato Chakraborty
University of Washington
Box 352500
Seattle, WA 98115
1-206-221-6513
swagato@u.washington.edu

Dipanjan Gope
University of Washington
1-408-765-3985
dips@u.washington.edu

Vikram Jandhyala
University of Washington
Box 352500
Seattle, WA 98115
1-206-543-2186
vj@u.washington.edu

ABSTRACT
Simulation of distributed electromagnetic effects of electrically large structures is no longer a luxury but a necessity in the accurate prediction of modern day circuit performance. In this regard, integral equation based methods have steadily gained in popularity but suffer from the time and memory bottlenecks arising from the resultant dense matrix. Fast linear complexity solvers have been introduced in the past but with the growing complexity of circuit layouts parallel implementations are the only viable options in addressing practical circuit layouts. In this paper, we present a parallel implementation of the low-rank compression based fast solver with linear cost reduction capacity with respect to the number of processors. The main problems in parallelizing a hierarchical algorithm are discussed and the advantages of the implemented scheme are highlighted. The new solver enables the simulation of full-chip problems consisting of millions of unknowns with acceptable accuracy and modest time and memory requirements.

Categories and Subject Descriptors
J.6 [**Computer-Aided Engineering**]: Computer-Aided Design

General Terms: Algorithms, Performance, Design

Key Words: Parasitics, Parallel, MPI, Compression

1. INTRODUCTION
Circuit theory, a low-frequency approximation to Maxwell's equations, has guided VLSI design and analysis for more than three decades. Electromagnetic simulation becomes necessary when the structure dimensions are comparable to the wavelength under consideration. Various electromagnetic simulation schemes have been developed to address this issue, as for example the Finite Difference Time Domain (FDTD) method, the Finite Element Method (FEM) and the Boundary Element Method (BEM). The latter is based on Maxwell's equations in integral form and has gained in popularity over the last few years owing the requirement of less number of unknowns to characterize the same problem.

Integral equation based methods however gives rise to a dense matrix the solution of which present a time and memory bottleneck. Various fast Krylov subspace based solver approaches, that exploit the physical properties of electromagnetic interactions captured through the Green's function, are available in existing literature. A few examples are the fast-multipole method (FMM) [1-2], adaptive integral method (AIM) [3], FFT based methods [4-5] and QR based algorithms [6-7]. These algorithms are capable of reducing the time and memory requirements to linear complexity with respect to the number of unknowns. However, when applied to real-life circuit layouts even the fast methods fall short of acceptable efficiency targets. Consequently, parallel implementations of fast algorithms are the most promising simulation strategy for handling practical circuit layouts. Moreover, the shift in the microprocessor road-map towards multiple core configurations implies the attractive possibility of parallel processing on user desktops.

In this work a parallel implementation of the fast QR based solver [7] has been implemented. The pre-determination of interaction list capability is exploited to reduce processor cross-communication requirements throughout the iterative process. In contrast, earlier QR techniques based on adaptive binary tree configurations, as well as fast multipole techniques requiring explicit multilevel tree traversal are significantly more complex to parallelize especially in terms of load balancing and minimizing inter-node communication.

2. INTEGRAL EQUATION FORMULATIONS

2.1 Capacitance extraction
To extract the system capacitance matrix, interfaces of conductor and dielectric bodies are divided into panels where basis functions describing the charge distribution are defined. A matrix equation representing the electrostatic interaction is set up after enforcing

Dipanjan Gope is now with Circuit Technology CAD, Intel Corp.

DAC 2006, July 24–28, 2006, San Francisco, California, USA.

appropriate boundary conditions on the conductor-to-dielectric and dielectric-to-dielectric interfaces. For detailed description, please refer to [7].

2.2 Inductance extraction

Inductance extraction is carried out through the computation of a complex frequency dependent impedance matrix of a multiport structure consisting of conductors. Conductors are modeled by the electric field integral equation assuming divergence-free electric current flowing on the surface of conductors and surface impedance model for resistive loss.

$$\mathbf{E}^S(\mathbf{J}) = -j\omega\frac{\mu}{4\pi}\int_S \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|}\mathbf{J}(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}ds' - Z_s\mathbf{J}(\mathbf{r}) \qquad (2.1)$$

Surface of conductors are discredited into triangles and to describe the current flow, RWG basis function are defined over each triangle pair, which share a common edge. Upon application of the boundary condition on the conductor surface, a matrix equation representing the "magnetostatic" (albeit with phase effects built into the Green's function) interaction is set up as

$$\begin{pmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1M} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2M} \\ \vdots & \vdots & \alpha_{ij} & \vdots \\ \alpha_{M1} & \alpha_{M2} & \dots & \alpha_{MM} \end{pmatrix}\begin{pmatrix} i_1 \\ i_2 \\ \vdots \\ i_M \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ V_{si} \\ 0 \end{pmatrix} \qquad (2.2)$$

where M is the total number of RWG edges, i_1, i_2 …. i_M are the unknown RWG basis function coefficients and V_s is the delta gap voltage excitation located at certain port i of the structure. After solving the matrix equation, the impedance matrix of the can be extracted from

$$\begin{pmatrix} Z_{11} & Z_{12} & \dots & Z_{1K} \\ Z_{21} & Z_{22} & \dots & Z_{2K} \\ \vdots & \vdots & Z_{ij} & \vdots \\ Z_{K1} & Z_{K2} & \dots & Z_{KK} \end{pmatrix}\begin{pmatrix} I_{p1} \\ I_{p2} \\ \vdots \\ I_{pK} \end{pmatrix} = \begin{pmatrix} V_{s1} \\ V_{s2} \\ \vdots \\ V_{sK} \end{pmatrix} \qquad (2.3)$$

where K is the total number of ports, I_{p1}, I_{p2} …. I_{pK} are the port currents and V_{si} is the delta gap voltage excitation located at certain port i of the structure.

3. PARALLEL MULTILEVEL LOW-RANK DECOMPOSITION ALGORITHM

The serial multilevel low-rank decomposition algorithm achieves efficient far-field sub-matrix compression through leveraging an oct-tree decomposition hierarchy and a merged interaction list. In this algorithm, the regions occupied by arbitrary shaped electromagnetic structures are hierarchically decomposed into small cubes and these cubes are represented by an Oct-tree data structure. The near field interactions are represented by a neighbor list at the finest level and far-field interactions are represented by interaction lists at all levels [7].

The overall computational workload for the multilevel low-rank decomposition algorithm includes the near-field Method of Moments (MoM) matrices set up, far-field interaction decomposition—Q and R low-rank matrix set up and matrix-vector multiplication. A scalable parallel algorithm requires balanced distribution of the above workload to different processors with minimized inter-processor communication. The iterative solution process is carried out through a Krylov Subspace iterative method that utilizes the fast matrix-vector products. In addition, a parallel preconditioner is needed for fast solution convergence and linear scaling. The presented algorithm fulfills above requirements through following steps.

3.1 Load balancing of near field interaction matrix construction and storage

As discussed above, near field interactions are represented by neighbor lists at the finest level and they can be grouped into a linked list data structure .

Each node denotes a neighbor interaction list and contains information on source and observer regions. To minimize the inter-processor communication, each processor builds its own copy of data structure that has a relatively trivial demand on both time and memory. As long as the workload associated with each node can be predetermined, the overall work load can be evenly distributed to each processor resulting in a scalable algorithm. By extracting the number of sources and number of observers at each node of the data structure, it is possible to find the exact load (modulo integer work at each processor) of near field MoM matrices set up for each processor as follows :

$$N_{work-load} = \sum_{i=1}^{h} \frac{m_i \times n_i}{np} \qquad (3.1)$$

where h is the total number of nodes, m_i is the number of observer, n_i is the number of source and np is the number of processors. Based on the predetermined work load, each processor holds similar amount of work $N_{work-load}$ as shown by figure 1:

MoM matrices representing near field interactions are constructed after the work load distribution as shown by figure 1:

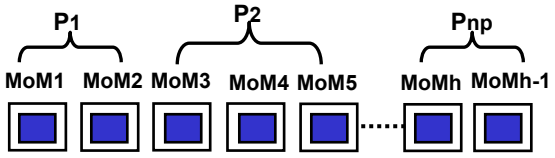

Figure 1. Near field interaction represented by MoM matrices

3.2 Load balancing of far field Q and R matrices set up and storage

In a manner similar to near field interactions, merged interaction lists [7] representing far field interaction across different levels can also be grouped into a link list.

Unlike the MoM matrices construction, it is not possible to predetermine the exact overall work load of far field interaction since the Q and R matrices have to be created before the exact rank of each node can be known. However, as discussed in [7], the oct-tree decomposition structure provides a predetermined rank-map for far field interaction sub-matrices across all the levels. A close estimate on each node's work load can be obtained

by utilizing the predetermined rank-map. Work load of far field Q and R matrices set up for each processor is derived as following:

$$F_{work-load} = \frac{\sum_{i=1}^{k} (m_i + n_i) \times r_i}{np} \qquad (3.2)$$

where k is the total number of nodes, m_i is the number of observer, n_i is the number of source, r_i is predetermined rank of certain type of merged interaction list and np is the number of processors. As shown by figure 2, after work load distribution, each processor creates its own Q and R matrices.

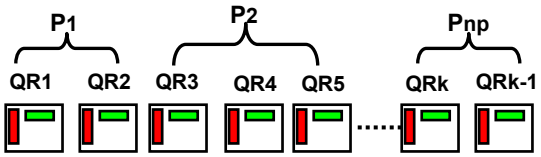

Figure 2. Far field interaction represented by QR matrices

3.3 Load balancing of matrix-vector multiplication

Given the fact that the work load of near and far field matrices set up on each processor is proportional to that of matrix-vector multiplication, load balancing scheme discussed above also applies to that of matrix-vector multiplication. As shown by figure 3, after near and far field matrices set up, each processor conducts its local matrix-vector multiplication, where **V** is the unknown vector.

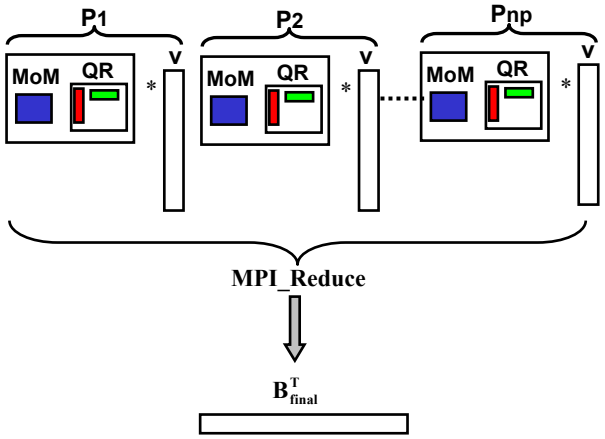

Figure 3. Matrix-vector multiplication and data collection

After matrix-vector multiplication, an MPI_Reduce operation is carried out to obtain the global matrix-vector product $\mathbf{B}_{final}$, that collects vector data from each processor.

3.4 Parallel sparse approximate inverse preconditioner

A scalable precondtioner based on the sparse approximate inverse algorithm (SPAI) is implemented and leads to significant speed-up on the iterative solve process. To construct the preconditioner, a sparse matrix **Z'** is extracted from the near field interaction and used to generate the preconditioner **M** for fast convergence.

$$\bar{\mathbf{M}}\bar{\mathbf{A}}\mathbf{x} = \bar{\mathbf{M}}\mathbf{b}$$

Figure 4 shows one example of the parallel SPAI preconditioner **M** drastically speeding up the iterative solving process.

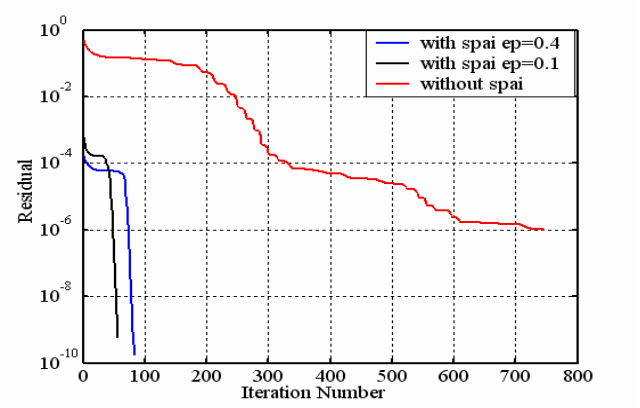

Figure 8. SPAI precontioner performance

In figure 4, ep denotes a threshold to control the quality of the preconditioner which functions as a efficiency-memory tradeoff

4. RESULTS

4.1 Capacitance extraction and efficiency test

All the simulations discussed in this section were run on a Linux cluster with total 18 nodes. Each node has two 3.2 GHz processors and 2G memory. For the purpose of validation and efficiency testing, capacitance extraction of one test structures was carried out using the proposed method..

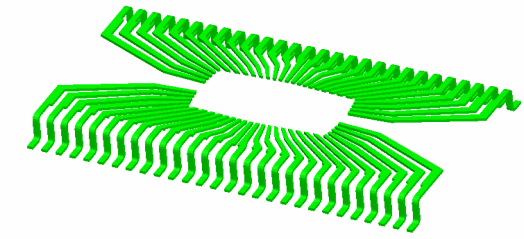

Figure 5. Packaging structure

The test structure , as shown in Figure 6, is one with ten copies of the 56-packaging-lead structure as shown in Figure 5. Good correlation of the extracted capacitance value with that from Ansoft's Q3D has been shown during the review of this paper.

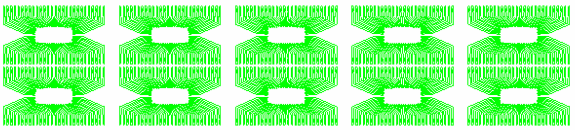

Figure 6. 560-packaging-lead structure

To test the efficiency of the load balancing schemes applied in the parallel algorithm, capacitance extraction of the second test structure was run on 1 to 17 processors. The number of unknowns on the test structure is 563,216. The test performance of the proposed method and assumed ideal performance were plotted in the same figure.

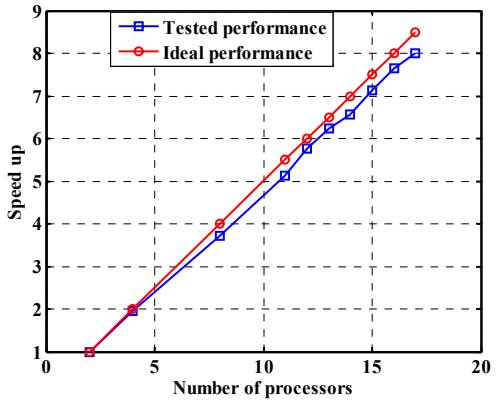

Figure 7. Simulation time vs. processor number

As shown by figure 7, the proposed method has linear scalability in both test cases owning to the balanced workload distribution.

4.2 Spiral inductor Q-factor extraction

The inductance extraction functionality of the proposed methodcan be used to extract the Q-factor of spiral inductors mostly appeared in RF circuit design. The simulation case is to extract the Q-factor of the spiral inductor which co-exists with

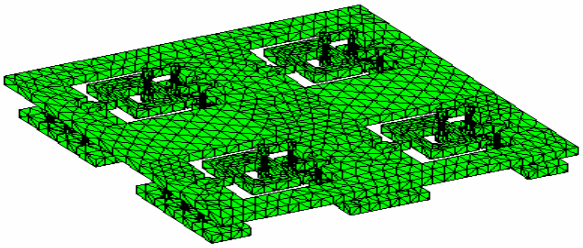

Figure 8. Co-planar spiral inductor

As shown by figure 9, the Q-factor of the co-planar case is lower than that of single inductor case due its mutual coupling including resistive and radiative losses to other three inductors.

Good correlation of single inductor's Q value with that from HFSS has been shown during the review process.

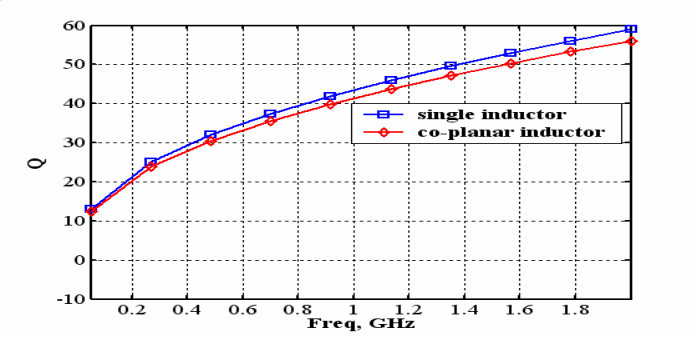

Figure 9. Single inductor Q-factor vs. co-planar inductor Q-factor

5. CONCLUSIONS

This paper presents a parallel multilevel low-rank decomposition algorithm for parasitic extraction of electrical large structures. Linear scalability is achieved through load balancing schemes on near and far field matrices construction. It is expected that the proposed scheme will enable large-scale distributed simulation due to the fact that both fast multilevel simulation technology and parallel processing are simultaneously exploited.

6. REFERENCES

[1] R. Coifman, V. Rokhlin and S. Wandzura, "The fast multipole method for the wave equation: a pedestrian prescription", *IEEE Trans. Antennas Propagat. Mag.,* vol. 35, pp. 7-12, June 1993.

[2] L.J. Jiang and W.C. Chew, "Modified fast inhomogeneous plane wave algorithm from low frequency to microwave frequency", *IEEE Antennas and Propag. Soc. Int. Symp.*, vol. 2 , pp. 22-27, June 2003.

[3] E. Bleszynski, M. Bleszynski and T. Jaroszewicz, "AIM: Adaptive Integral Method for Solving large-scale electromagnetic scattering and radiation problems", *Radio Science*, vol. 31, pp. 1225-1251, Sept-Oct 1996.

[4] N. Yuan, T.S. Yeo, X.C. Nie and L.W. Li, "A Fast Analysis of Scattering and Radiation of Large Microstrip Antenna Arrays", *IEEE Trans. on Antennas and Propagation*, vol. 51, pp. 2218-2226, Sep 2003.

[5] Z. Zhu, B. Song and J. White, "Algorithms in FastImp: a fast and wide-band impedance extraction program for complicated 3-D geometries", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, pp. 981 – 998, July 2005

[6] S.Kapur and D.Long, "IES^3: a fast integral equation solver for efficient 3-dimensional extraction", *International conference on Computer Aided Design*, pp. 448-455, Nov. 1997.

[7] D.Gope and V.Jandhyala, "Oct-Tree Based Multilevel Low-Rank Decomposition Algorithm for Rapid 3D Parasitic Extraction", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol-23, pp. 1575 - 1580, Nov. 2004.