

Enhancing the Performance of Electromagnetic Applications on Clustered Architectures

Maryam Mehri Dehnavi and Dennis D. Giannacopoulos

Department of Electrical and Computer Engineering, McGill University, Montreal, QC H3A 2A7, Canada

Scalability and low power consumption are major issues in designing custom hardware platforms for accelerating computation intensive electromagnetic applications. Clustered architectures decreases wire delays, power consumption and enhance scalability. We introduce a new steering algorithm that increases the performance of electromagnetic applications on such architectures, making clustering an even more attractive design alternative. The effectiveness of the new method is illustrated for two representative electromagnetic benchmarks.

Index Terms—Clustered architectures, computational electromagnetics, hardware acceleration.

I. INTRODUCTION

THE computation intensity of electromagnetic (EM) applications increases as technology advances, thus necessitating the design of scalable hardware. Unfortunately scaling centralized and general purpose hardware with the application, increases the complexity and delays on the architecture components as well as design cost and power consumption.

Clustering of computational elements is becoming widely recognized in EM simulation as an effective method to overcome some of the scaling, complexity, and power problems [1], [2]. Because the design effort of producing a single cluster is greatly simplified, scalability is enhanced in such architectures [1]. In other words, as the workload of electromagnetic applications increase, more of the cluster cores can be added to the processor for low cost design. The distribution of architecture resources amongst clusters increase heat dissipation and reduce power consumption in clustered processors, and the reduced wire length between the resources in each of the clusters enable higher operating clock frequencies for each cluster in such architectures.

One of the major disadvantages of clustered architectures is their reduced speedup compared to monolithic designs with similar resources. The delays caused due to inter-cluster communications, as well the load imbalance, reduces the throughput of EM applications on such architectures. Thus, using clustered hardware platforms for electromagnetic applications implies a tradeoff of performance for scalability, low cost and power consumption.

Techniques to improve the throughput of EM applications on clustered architecture can be categorized as hardware (dynamic) and software (static) optimizations. Software optimization methods designed for EM applications are performed either on or before compile time and, unfortunately, lack runtime feedback [2]. Hardware optimization techniques, on the other

hand, tend to maximize the execution speed of an application by using runtime information. The large dependency chains in electromagnetic applications lead to unpredictable behavior for these applications at runtime; thus, dynamic optimization techniques that use runtime information are suitable for enhancing the performance of EM application on clustered architectures.

Instruction steering heuristics [3] are popular hardware optimization methods used to accelerate the execution speed of EM applications on clustered processors. Such heuristics are designed to distribute instructions between clusters in a clustered architecture. Efficient steering topologies maximize performance by minimizing inter-cluster communication delays while maintaining load balance. The purpose of this paper is to introduce a new steering algorithm that increases the performance of computationally intensive electromagnetic applications on clustered architectures.

This paper is organized as follows. In Section II, we briefly described the two electromagnetic applications evaluated on the clustered architecture. Section III studies the performance that can ideally be achieved from a clustered architecture. The efficiency of the Seek Dependency steering algorithm proposed in previous work on enhancing electromagnetic applications is studied in Section IV. In Section V, we propose the new Switching Thresholds steering topology and evaluate its ability to enhance the execution of electromagnetic applications on clustered architecture. The clustered architecture used in the work is presented in Section VI. Finally, conclusion and suggestions for future work are presented in Section VII.

II. BENCHMARK SPECIFICATIONS

In this section, we introduce the two electromagnetic applications used in this work to study the efficiency of steering algorithms in enhancing the execution speed of EM applications on clustered architecture.

The *GemsFDTD* benchmark [4] from the SPECCPU2006 benchmark suite solves the Maxwell's equations in 3-D in the time domain using the finite-difference time-domain (FDTD) method. The radar cross section (RCS) of a perfectly conducting (PEC) object is computed. At the core of the FDTD method are second-order accurate central-difference approximations of the Helmholtz equation. These central-differences are employed on a staggered Cartesian grid resulting in an explicit finite-difference method. An incident plane wave is generated using

Manuscript received October 07, 2008. Current version published February 19, 2009. Corresponding author: D. D. Giannacopoulos (e-mail: dennis.giannacopoulos@mcgill.ca).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TMAG.2009.2012620

TABLE I
INPUT PARAMETERS OF PEC SPHERE USED FOR GEMSFDTD BENCHMARK

Parameter	Value
Problemsize	210,210,210
Cellsize	0.01,0.01,0.01
Radius	0.205
Number of steps	1000
Planewave	Gaussian
Outerboundary	Upmlhigh
CFL	0.95
Iteration	100

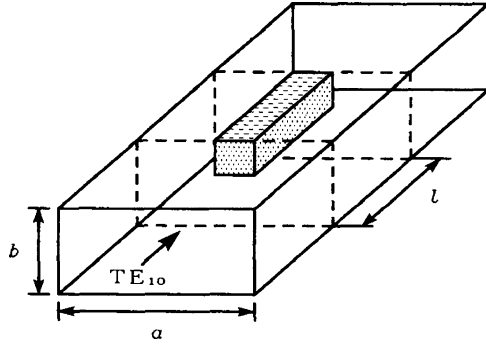


Fig. 1. E -plane ridge waveguide with discontinuity used for *deal.II* benchmark [6]. Note: $a = 19.05$ mm; $b = 9.524$ mm; $l = 5.08$ mm; discontinuity width = 1.016 mm; discontinuity height = 7.619 mm [5].

Huygens' surfaces. The computational domain is truncated by an absorbing layer in order to minimize the artificial reflections at the boundary. The uniaxial perfectly matched layer (UPML) is used and a time-domain near-to-far-field transformation computes the RCS. We have used a PEC sphere with the input parameters indicated in Table I as our input to the *GemsFDTD* benchmark.

The SPEC CPU2006 benchmark 447.*deal.II* is a program that uses *deal.II*, a C++ program library targeted at adaptive finite elements and error estimation. The main aim of *deal.II* is to enable development of modern finite element algorithms, using among other aspects sophisticated error estimators and adaptive meshes. The code uses modern adaptive methods based on duality weighted error estimates to generate optimal meshes. The equations are solved in 3-D. The input we have used is a problem of wave scattering by an obstacle ($\epsilon_r = 6$) in a rectangular waveguide, which is illustrated in Fig. 1. The problem is simulated using brick edge elements for the electric-field calculations [5]. Fig. 1 also specifies the geometric values used in the problem.

III. IDEAL SPEEDUP IN CLUSTERED ARCHITECTURE

An ideal clustered architecture with an ideal steering topology would have no inter-cluster communication delays and the work load would be completely balanced amongst its clusters. To prove the importance of research on designing efficient steering heuristics, we have simulated the ideal clustered architecture and steering topology for EM applications. Such an assumption is not approachable in reality because of existing wire delays in the underlying network topology of clustered

TABLE II
IDEAL CLUSTERING SPEEDUP FOR EM BENCHMARKS

Benchmark	GemsFDTD	deal.II	Average
% Speedup Ideal Clustering	102.5	104	103.25

The maximum speedup achieved from an ideal steering heuristic on a clustered architecture with reference to the non clustered architecture.

architecture and the considerable dependencies between the instructions of electromagnetic applications.

The purpose of studying the performance of EM applications on an ideal clustered architecture is to show current steering algorithms have the potential to be further enhanced to accelerate the execution speed of such applications. To simulate such a heuristic, we have to set the network link delay to zero and regardless of the dependency between the instructions, distribute them between clusters in a round-robin fashion.

To measure speedup we have assumed the performance gained from the non clustered architecture as a reference and used the following formula to calculate speedup throughout the paper:

$$\text{Speedup} = 100 \times \frac{\text{InstructionsPerCycle_clustered}}{\text{InstructionPerCycle_nonclustered}}. \quad (1)$$

As shown in Table II, the performance achieved from the ideal case is almost the same as the non-clustered architecture for the EM applications. We have to consider that the main purpose of clustering hardware platforms for electromagnetic applications is to benefit from the scalability, low power consumption and low design cost of clustered architecture.

In summary, enhances in the instruction steering algorithms for clustered architecture would ideally increase their performance for EM applications. Thus the execution speed of electromagnetic applications on a clustered architecture with an ideal steering heuristic would potentially be very close to the non-clustered version while consuming less power and being cheaper to design.

IV. SEEK DEPENDENCY STEERING ALGORITHM

Various instruction steering topologies have been introduced in previous work. Among them seek dependency (SDEP) [1] has proven to perform the best for computation intensive applications. In this section, we will study the efficiency of this algorithm in enhancing the execution speed of EM applications. The SDEP steering heuristic steers instructions based on the following two strategies: minimizing communication delays and maximizing load balance.

If instruction I1 produces the input operands of instruction I2, then I1 and I2 are called the producer and the consumer instructions respectively. In order to decide which cluster to steer an instruction to, SDEP will first determine if any cluster holds a producer to the current instruction. If so, because SDEP gives priority to minimizing communication delays, the instruction will be steered to that cluster. If the instruction has no producer in the clusters or the cluster holding its producer instruction is full, to preserve load balance SDEP will tend to steer the instruction to the least loaded cluster.

TABLE III
SPEEDUP FOR THE SDEP STEERING HEURISTIC

Benchmark	GemsFDTD	deal.II	Average
%Speedup SDEP	43.3	39.2	41.25

Speedup achieved from the SDEP steering heuristic on a clustered architecture with reference to the non-clustered processor.

TABLE IV
PERCENTAGE OF INTER-CLUSTER COMMUNICATIONS IN SDEP

Percentage of communication between clusters	SDEP with communication reduction	SDEP without communication reduction
GemsFDTD	10.3	56.2
deal.II	15.6	67
Average	12.95	61.6

percentage of executed instructions from each cluster

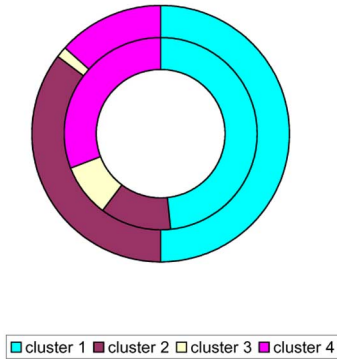


Fig. 2. Load distribution amongst clusters in SDEP.

TABLE V
PERCENTAGE OF RESOURCES IDLE USING SDEP

Percentage of time resources are idle	Cluster one	Cluster two	Cluster three	Cluster four
GemsFDTD	10	77.2	80	35.5
deal.II	12	22	89	23
Average	11	49.6	84.5	29.95

Table III shows the performance of the SDEP technique for the *GemsFDTD* and *deal.II* electromagnetic applications. As demonstrated in the table, the heuristic is only capable of maintaining on average 41% of the lost performance due to clustering. Although the technique is able to reduce the number of inter-cluster communication delays by 48% compared to when there is no communication reduction strategy (see Table IV), but because of the large dependency chains in EM applications, instructions are mainly steered to specific clusters. As shown in Fig. 2, this will result in an uneven distribution of load amongst clusters and a poor speedup for EM applications using SDEP. As a result the resources of the least loaded clusters would be idle most of the time (see Table V) when using SDEP.

TABLE VI
SPEEDUP FOR THE ST STEERING HEURISTIC AND BEST THRESHOLDS

BENCHMARK	GemsFDTD	deal.II
Best Upper Threshold	0.06	0.052
Best Lower Threshold	0.02	0.03
% SPEEDUP ST	88.3	91.2

Speedup achieved from the ST steering heuristic on a clustered architecture with reference to the non clustered processor.

These results indicate that the large dependency chains in electromagnetic applications reduce the ability of steering topologies such as SDEP in minimizing communication delays; thus, make steering topologies such as SDEP unsuitable for increasing their execution speed on clustered architectures.

V. SWITCHING THRESHOLDS STEERING TOPOLOGY

As discussed in Section IV, existing steering topologies such as SDEP have a poor performance for EM applications. Such techniques loose the control over load balance while trying to minimize inter-cluster communication delays for EM application with long dependency chains. The switching thresholds (ST) steering topology proposed in this section uses the same strategy as SDEP to minimize inter-cluster communication delays. To overcome the load balance issue of the SDEP heuristics in the ST topology, we define a load balancing factor to measure load distribution amongst clusters. The factor is determined at each step by calculating the coefficient of variation (standard deviation/mean) for the distributed load among clusters. If this factor exceeds an upper threshold (UT), the steering logic will switch from SDEP to another steering topology namely, the least cluster load balancing (LCLB) technique. LCLB steers instructions to the least load clusters and will remain as the steering logic until the load balance factor is reduced to the lower threshold (LT). Unlike the dependence base with load balancing technique proposed in [1] and [3], the upper and lower thresholds in the ST steering logic are dynamically determined based on the performance of the EM applications the clustered architecture. Thus, the upper and lower thresholds differ for each application allowing the benchmark benefit the most from switching between the LCLB and SDEP algorithms.

The main purpose of switching from SDEP to LCLB is to resume load balance between clusters. Thus, the main idea behind the ST steering heuristic is to use SDEP to minimize communication delays and use the LCLB method to balance load amongst clusters by switching between these heuristics. We have measured the performance of the *GemsFDTD* and *deal.II* electromagnetic applications using various upper and lower thresholds and have presented the thresholds that result in maximum performance for these applications in Table VI. The table also shows the maximum speedup achieved from these thresholds. The results prove that the ST steering heuristic is capable of maintaining an average 91% of the lost performance due to clustering for EM applications. The new method outperforms old steering algorithms such as SDEP up to 50% for *GemsFDTD* and *deal.II* electromagnetic benchmarks.

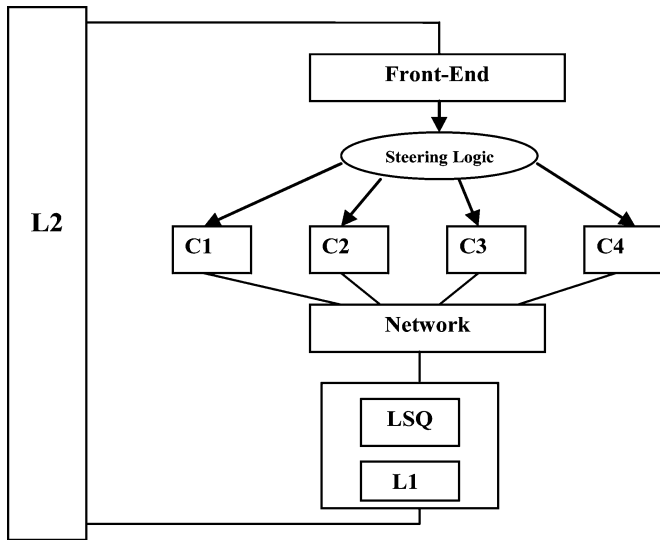


Fig. 3. Micro-architecture of our four clustered processor (C1-4: each of the clusters. LSQ: Load-Store Queue. L1-L2: data and instruction caches).

Hence, benefiting from both the SDEP and LCLB steering heuristics, the ST steering topology is very efficient in enhancing the execution speed of electromagnetic applications that have large dependency chains.

VI. CLUSTERED ARCHITECTURE TESTBED

To study the performance of EM application on clustered architecture, we have simulated the clustered processor presented in Fig. 3 by implementing clustering to the SimpleScalar [6] simulator. The front-end of the architecture consists of the fetch queue and decode stage and is kept centralized while the back-end consisting of the register rename unit, register files and functional units are clustered. A bidirectional ring topology is used as the underlying network to preserve communication among clusters [7]. Each hop in the network is equal to a one cycle delay in the simulator.

The two benchmarks used as representative EM applications are *GemsFDTD* and *deal.II* from SPEC CPU2006 [4] benchmark suites, as noted above.

VII. CONCLUSION

In this paper, we studied the performance of the SDEP steering heuristic in enhancing the execution speed of EM applications by running the *GemsFDTD* and *deal.II* electromagnetic applications on a four clustered architecture. We demonstrated that the long dependency chains in such applications reduce the efficiency of SDEP. The poor load balance and idle resources in the clusters reduced the execution speed of EM applications on clustered architecture. To overcome the load imbalance in SDEP, we proposed the switching thresholds steering logic. The results showed that for EM applications, the switching thresholds heuristic outperforms SDEP by 50%, and is capable of maintaining an average 91% of the lost performance due to clustering.

As future work, we intend to study the simultaneous execution of EM applications on clustered architectures and how to enhance their execution speed on such hardware.

ACKNOWLEDGMENT

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

- [1] A. Aggarwal and M. Franklin, "An empirical study of the scalability aspects of instruction distribution algorithms for clustered processors," in *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw. (ISPASS)*, Nov. 2001, pp. 172–179.
- [2] R. Balasubramanian, S. Dwarkadas, and D. Albonesi, "Dynamically managing the communication-parallelism trade-off in future clustered processors," in *Proc. 28th Int. Symp. Comput. Arch.*, Dec. 2005, pp. 275–286.
- [3] R. Canal, J. M. Parcerisa, and A. González, "Dynamic code partitioning for clustered processors," *Int. J. Parallel Program.*, pp. 59–79, Feb. 2001.
- [4] Standard Performance Evaluation Corporation, *SPEC Newsletter*, Sep. 2006.
- [5] K. Ise, K. Inoue, and M. Koshiba, "Three dimensional finite element method with edge elements for electromagnetic waveguide discontinuities," *IEEE Trans. Microw. Theory Technol.*, vol. 39, no. 8, pp. 1289–1295, Aug. 1991.
- [6] D. Burger, T. M. Austin, and S. Bennett, "Evaluating future microprocessors: The SimpleScalar tool set," Univ. Wisconsin-Madison, Madison, Tech. Rep. CS-TR-96-1308, 1996.
- [7] J. Parcerisa, J. Sahuquillo, A. González, and J. Duato, "On-chip interconnects and instruction steering schemes for clustered microarchitectures," *IEEE Trans. Parallel Distrib. Syst.*, vol. 16, no. 2, pp. 130–144, Feb. 2005.