Efficient Markov Clustering Model for VLSI Circuit Partitioning

Article *in* Australian Journal of Basic and Applied Sciences · October 2012

CITATION
1

READS
53

4 authors, including:

R. MANIKANDAN
SASTRA University
44 PUBLICATIONS **42** CITATIONS
SEE PROFILE

Sujitha Rajendiran
Tata Consultancy Services Limited
4 PUBLICATIONS **1** CITATION
SEE PROFILE

Australian Journal of Basic and Applied Sciences, 6(10): 31-35, 2012
ISSN 1991-8178

Efficient Markov Clustering Model for VLSI Circuit Partitioning

[1]R. Manikandan, [2]S. Sathvi, [2]C. Thenmozhi, [3]R. Sujitha

[1]Senior Asst Prof, School of Computing, SASTRA University, Thanjavur-613401, India.
[2]Department of ICT, School of Computing, SASTRA University, Thanjavur-613401, India.
[3]M.Tech VLSI Design, School of Computing, SASTRA University, Thanjavur-613401, India.

Abstract: Partitioning is a significant area of VLSI Design and has been an energetic area of research. VLSI circuit partitioning is an imperative part of physical design stage. In this paper, we compare the performance of K-way partitioning and Markov Clustering Algorithm. This comparative study will provide a clarification to approach problems in circuit partitioning pertaining to VLSI Design. The performance of both approaches is compared using Simple Full Adder Circuit. From this analysis it is proved that the Markov Clustering Algorithm achieves greater performance compared to K-way partitioning

Key words: Circuit partitioning, Hypergraph, K-way Partitioning, Markov Chain Clustering,.

INTRODUCTION

Partitioning is the process of dividing the design into smaller, more manageable modules. The objective is to divide the circuit into blocks such that each component falls within the prescribed sizes and the complexity of connection between the components is reduced (George Karypis and Vipin Kumar, 2000). Divide-and-conquer methods including partition based and cluster-based algorithms have recently been developed to avoid long runtimes. A comprehensive survey of partitioning classified most algorithms into FM and KL families. Clustering recognizes natural groups within a class of entities. With the recent advances in VLSI technology, clustering is being widely used to identify natural circuit hierarchies to enable the divide and conquer design methodology. Clustering effectively reduces the problem size and guides the partitioning process to improve the results. (K. A. Sumitra Devi, N. P. Banashree, and Annamma Abraham, 2007).

This paper discusses two methods such as K-way partitioning and Markov Clustering Algorithm. The K-way partitioning problem is one of the methods to partition a network into K subsets (partition) of approximately the same size while minimizing the number of interconnections between the K partitions. Whereas Markov Clustering algorithm utilizes only the most recent past values for the results to be produced.

K-Way Partioning:

This problem has many applications in VLSI circuit Design ranging from Circuit layout to logic simulation and also in many fields like distributed/parallel computing, packet-switched networks, data mining and memory management.

Formally, a hypergraph G=(V,E) is defined as a set of vertices V and a set off hyperedges E, where each hyperedge is a subset of the vertex set V. The K-way hypergraph partitioning problem is defined as follows:

Consider a graph G=(V,E) and an overall load balance tolerance b such that 0<b<1, the goal is to partition the set V into k subsets, $V_1,V_2,\ldots V_k$, for k≥1.0, such that the number of vertices and a function defined over the hyperedges are optimized. The corresponding weights are given by (R.Manikandan, Jaladhanki Sindura, M.D.L. SriRavali and P.Swaminathan ,2012) (M.Thiyagarajan and R.Manikandan,2011).

$W_i = fw(P_i)$, $1 \leq i \leq k$,

such that:

$$W_i < (1+\varepsilon)Wavg$$

Holds for all $1 \leq i \leq k$ (where Wavg =∑ Wi/ k).

Procedure:

Step 1: The number of vertices in the hypergraph is given as input 'N'.
Step 2: The respective adjacent nodes for each vertex will form an adjacency matrix and their weights, Wi.
Step 3: Consider a balance factor b, where 0<b<1 and input the number of partitions to be divided, k.
Step 4: Calculate the average weight of all the nodes, Wavg.
Step 5: The adjacent nodes are classified into a partition till they satisfy the condition,
$W_i<(1 + b) W_{avg}$, and rest are classified into other partitions based on the same condition.

Corresponding Author: R. Manikandan, [1]Senior Asst Prof, School of Computing, SASTRA University, Thanjavur-613401, India.
E-mail: manikandan75@core.sastra.edu,

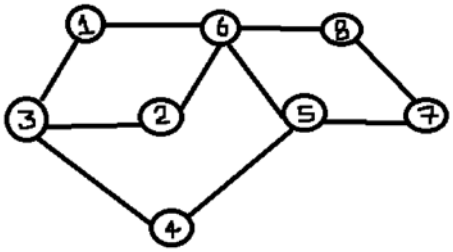


Fig. 1: Let us consider the above hypergraph (fig-1) with 8 vertices that represents a Simple Full Adder Circuit.

Vertices	Adjacent Nodes
1	2,7
2	1,3,5,6
3	2,4
4	3,5
5	2,4,8
6	2,7
7	1,6,8
8	5,7

Weights of the vertices are considered from 0 to 7.
The Average weight (Wavg) is 3
The number of partitions is considered to be K=2 and the balance factor is b=0.9.
The partition result is,
Partition 1: 1, 4, 2
Partition 2: 3, 5, 6, 7, 8

Markov Clustering Algorithm:

The Markov Clustering algorithm (MCL) is an unsupervised clustering method for graphs. Markov processes are independent of more distant values; it depends only on the most recent past values. The graphs may be weighted with nonnegative weights.(DAI Hui, ZHOU Qiang, BIAN Jinian, 2011).

Let G be the input graph. MCL simulates the flow in G by first identifying G in a canonical way with Markov graph G1. The flow is alternatively expanded and inflated, leading to a row of Markov graphs Gi. The expansion and inflation parameters of the MCL process influence the output granularity. MCL is space-efficient, time-efficient and lends itself to large scaling.MCL performs inflation operation, by changing the transition values for each vertex, so as to strengthen the strong neighbours and demote the less popular neighbours.

Given a Matrix M and a real nonnegative number r, the matrix resulting from rescaling each of the columns of M with power coefficient r is called Ґr M and Ґr is called inflation operator with power coefficient r, formally the expression is defined by (Manikandan, R., 2012).

$$(\text{Ґr Mpq}) = (\text{Mpq})^r / \sum_{i=1}^{k} (\text{Miq})^r$$

MCL also performs Expansion operation, i.e., taking the Markov chain transition matrix powers. The Expansion operator is responsible for allowing flow to connect different regions of the graph.

Procedure:
Let G (V, E) is an input graph (fig-1).
Step 1: graph is given as input
Step 2: Inflation parameter r and power parameter e are given as input by user (say 2).
Step 3: create the Association matrix for the graph G, (say MG)

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Step 4: self loop for each node is created.

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Step 5: Create a transition matrix (say TG).

$$\begin{pmatrix} 0.3 & 0 & 0.25 & 0 & 0 & 0.2 & 0 & 0 \\ 0 & 0.3 & 0.25 & 0 & 0 & 0.2 & 0 & 0 \\ 0.3 & 0.3 & 0.25 & 0.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.25 & 0.3 & 0.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.3 & 0.4 & 0.2 & 0.3 & 0 \\ 0.3 & 0.3 & 0 & 0 & 0.4 & 0.2 & 0 & 0.3 \\ 0 & 0 & 0 & 0 & 0.4 & 0 & 0.3 & 0.3 \\ 0 & 0 & 0 & 0 & 0 & 0.2 & 0.3 & 0.3 \end{pmatrix}$$

Step 6: Expand Matrix by taking the eth power (Here e=2).

$$\begin{pmatrix} 0.225 & 0.135 & 0.137 & 0.075 & 0.080 & 0.10 & 0 & 0.060 \\ 0.135 & 0.225 & 0.137 & 0.075 & 0.080 & 0.10 & 0 & 0.060 \\ 0.165 & 0.165 & 0.2875 & 0.165 & 0.12 & 0.12 & 0 & 0 \\ 0.075 & 0.075 & 0.1375 & 0.285 & 0.28 & 0.08 & 0.12 & 0 \\ 0.060 & 0.060 & 0.075 & 0.21 & 0.48 & 0.12 & 0.21 & 0.15 \\ 0.15 & 0.15 & 0.15 & 0.12 & 0.24 & 0.3 & 0.21 & 0.15 \\ 0 & 0 & 0 & 0.12 & 0.28 & 0.14 & 0.3 & 0.18 \\ 0.060 & 0.060 & 0 & 0 & 0.2 & 0.1 & 0.18 & 0.24 \end{pmatrix}$$

Step 7: Inflation of Matrix by r (Here r=2).

$$\begin{pmatrix} 0.415 & 0.097 & 0.0813 & 0.031 & 0.005 & 0.057 & 0 & 0.025 \\ 0.132 & 0.269 & 0.0813 & 0.031 & 0.005 & 0.057 & 0 & 0.025 \\ 0.197 & 0.144 & 0.357 & 0.152 & 0.012 & 0.082 & 0 & 0 \\ 0.040 & 0.029 & 0.813 & 0.455 & 0.064 & 0.036 & 0.034 & 0 \\ 0.026 & 0.019 & 0.024 & 0.247 & 0.189 & 0.082 & 0.105 & 0.158 \\ 0.162 & 0.119 & 0.097 & 0.080 & 0.047 & 0.514 & 0.105 & 0.158 \\ 0 & 0 & 0 & 0.080 & 0.064 & 0.112 & 0.214 & 0.227 \\ 0.162 & 0.319 & 0 & 0 & 0.032 & 0.057 & 0.540 & 0.405 \end{pmatrix}$$

Step 8: Repeat Steps 6 and 7, to reach a Steady State (Convergence).

$$\begin{pmatrix} 1.0 & - & 1.0 & - & - & 1.0 & - & - \\ - & 1.0 & 1.0 & - & - & 1.0 & - & - \\ 1.0 & - & 1.0 & - & - & - & 1.0 & - \\ - & 1.0 & - & - & 1.0 & - & - & - \\ 1.0 & - & 1.0 & - & - & 1.0 & - & - \\ 1.0 & - & 1.0 & - & - & - & 1.0 & - \\ - & 1.0 & - & - & 1.0 & - & - & - \\ - & - & - & - & 1.0 & - & - & - \end{pmatrix}$$

It is not obvious that the result will converge. However it shown experimentally that it often does occur [5].

Performance Analysis:

Simple Full Adder Circuit:

The Full-Adder is arithmetic Circuit (fig-2) Capable of performing addition on three single-bit numbers. It has three input ports (A, B, Cin) and two Output ports (Sum, Cout).

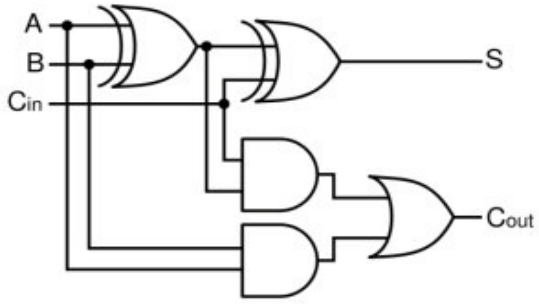
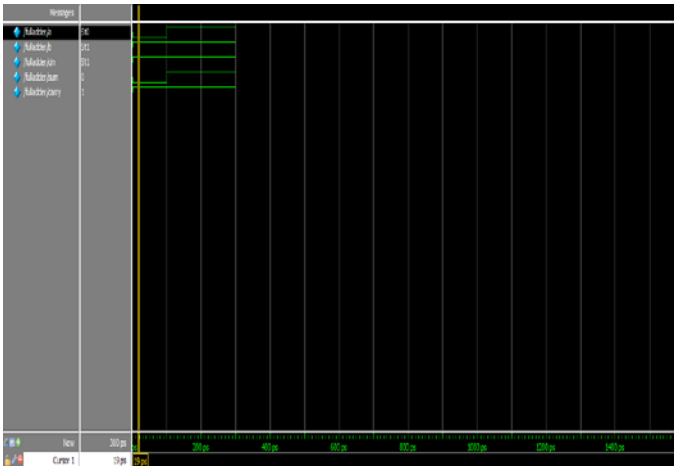


Fig. 2: The Above Circuit is represented as graph.

(fig-1)and given as input to K-way partitioning and Markov Clustering Algorithms.

The Result for the Full-Adder Circuit implementation is shown below.



Observation and Result:

From the observation of the result of Full-Adder Circuit, we can infer that k-way partitioning considers weight of the nodes based on which the partitioning is done. Only by adjusting the balance factor we can get more optimized results of partitioning. Whereas Markov Clustering algorithm is very fast, scalable unsupervised clustering method based on simulation of stochastic flow in graph. Thus the performance of Markov Clustering algorithm is more efficient in VLSI Circuit partitioning when compared to k-way partitioning algorithm.

Conclusion:

In this paper, we have discussed and evaluated the performance of algorithms like k-way partitioning method and Markov Clustering method for the process of VLSI partitioning circuit design. By the evaluation, we conclude that Markov Clustering Algorithm gives better result, It takes less time for the partitioning of circuit because the probability of the next Time step depends only on the current probabilities and hence it is independent of distant value.

REFERENCES

DAI Hui, ZHOU Qiang, BIAN Jinian, 2011. ”Markov Clustering-Based Placement Algorithm for Hierarchical FPGAs”, Tsinghua Science And Technology ISSN 1007-0214 10/17, 16: 62-68.

George Karypis and Vipin Kumar, 2000. ”Multilevel k-way Hypergraph Partitioning”, VLSI Design, 11(3): 285-300.

K.A. Sumitra Devi, N.P. Banashree and Annamma Abraham, 2007. ”Comparative Study of Evolutionary Model and Clustering Methods in Circuit Partitioning Pertaining to VLSI Design”, World Academy of Science, Engineering and Technology.

Manikandan, R., Jaladhanki Sindura, M.D.L. SriRavali and P. Swaminathan, 2012. “Efficient mathematical Model on VLSI Circuit Partitioning”, Research Journal of Applied Sciences, Engineering and Technology, 4(13): 1978-1979.

Thiyagarajan, M. and R. Manikandan, 2011. “Hypergraph and Coprocessor design for VLSI Partitioning Problems”, Journal of Theoretical and Applied Information Technology, 26(2): 107-111.