Contents list available at IJMEC

IJMEC

International Journal of Mechatronics, Electrical and Computer Technology (IJMEC)

Journal Homepage: www.aeuso.org

PISSN: 2411-6173 - EISSN: 2305-0543

Investigation and Evaluation of Program Code Optimization Algorithms for Reducing Power Consumption by Combining Genetic and K-means Algorithms

Sajad Bahrekazemi and Asadollah Shahbahrami

Master of Computer Engineering, University of Guilan, Guilan, Iran

Associate Professor, Department of Computer Engineering, University of Guilan, Guilan, Iran

Corresponding Author's E-mail: sbahrekazemi@ymail.com

Abstract

Since 1990, power has been one of the major restrictions in computer systems. Due to lower energy reserves, the importance of saving power is increasingly felt. Today this issue is much more evident in computer science, especially computer programs. Further savings in power is more acceptable in different software aspects because power saving in software aspects is more cost effective than the hardware aspects. Power consumption is a major challenge in programs, thus code optimization algorithms accelerate application's navigation and optimizes reduced power consumption. In the applications, by reducing the consumed power, the processor will have free time and its waste of time is prevented. For example, when a program runs in a CPU with slow clock, it consumes less energy than is the case when it runs with higher clock speed. The purpose of this paper is to reduce power consumption by applications. The power consumption has become a major challenge in computer systems and knowledge and access to parts of the program that have higher power consumption is very important because, by having access to these parts, it is possible to examine their improvement methods. Here, genetic and K-means algorithms are combined and an algorithm is proposed for code optimization that uses the loop interchange optimization technique which can reduce the power consumption by the program. The implementation results indicate the reduced power consumption in implementation of various matrix sizes. Program processing. The purpose of code optimization algorithm is to improve the performance of programs to reduce their power consumption.

Keywords: *Code optimization algorithms, reducing power consumption, genetic algorithm, K-means algorithm.*

I. INTRODUCTION

In recent decade power consumption has become a major issue for the computer systems [15]. In portable systems such as laptops, mobile devices and so on low power consumption is required. Even in cases where there is no power problem, low power consumption is a requirement for example in the microprocessors that have high-volume operations, problems such as expensive cooling systems are evident. This is issue clear in the computer programs and it is considered as an important feature. Thus, the power consumption plays an important role in computer programs and causes the parts of program with higher power consumption to be assessed more [18]. For example, during running the program in CPU as running the program is longer, CPU is more engaged and more energy is consumed; therefore, the role of power consumption becomes more prominent. Moreover the style of programming can affect power consumption or in the multiplexing technique in which data Fourier transform is done as parallel and simultaneously, the addition and multiplication calculations are

Article History:
Received Date: Oct. 08, 2018
Accepted Date: Dec. 06, 2018
Available Online: Jan. 04, 2019

reduced which in turn leads to a huge reduction in power consumption. The aim of optimization is to reduce the size, power and energy consumption in applications [2]. Although the techniques, methods and multiple algorithms have been proposed to reduce the power consumption, it is necessary to use intelligent techniques to find an appropriate approximation solution for computer programs. Using compiler optimization methods significant improvements are obtained in computer programs. To achieve the desired result choosing the appropriate methods is very important.

II. STATEMENT OF THE PROBLEM

In recent years, the main focus was on improving performance and techniques regardless of the power. In conferences and seminars the concept of power was considered as a marginal issue but over time the concept of power consumption became important because of the importance of saving power and energy in computer systems and avoiding wasting power and energy and it was observed in a variety of architectures and computer systems. Today power cannot be ignored in any architecture [15]. Due to the increase in portable systems, power consumption is of particular importance and has become one of the problems of today's systems, thus reducing it improves system performance greatly. Code optimization is also an important part of computer programs that aims to improve the performance and has many applications in software systems; which is followed by reduced power consumption. This problem is important in computer programs such that in 2011 and 2012 in the USA the power of computers was reduced into 10% [3, 4]. In the late 1990s, power was known as the first order system design constraint by the system designers around the world. Thus, power cannot be ignored [15]. Therefore the methods and ways to reduce power consumption should be reviewed and evaluated. The programs’ code optimization is one of these solutions that should be studied seriously.

III. RESEARCH METHOD

In this paper genetic and K-means algorithms are combined and an algorithm is proposed for code optimization that using loop interchange optimization technique could reduce the power consumed by an application (matrix n * n) in the implementation of various matrix sizes. The reason to choose this program is that the loops are the best option for optimization, because the highest time is spent on the loops [16]. Loop optimization has particular importance in accelerating the implementation of programs because the code usually runs within a loop several times; thus the main goal of this research is to reduce power consumption.

```
for (x = 1; x < n; x ++)
   for (y = 1; y < n; y ++)
     a[x][y] = a[x-1][y] + a[x][y-1]
```

A. *Genetic*

Genetic algorithm is among the evolutionary algorithms that uses the genetic and mutation techniques. This algorithm is part of the developments in computer science introduced by John Holland in 1970 year and it is a search technique in computer science to find approximate solutions in optimization and search problems [12, 10]. This algorithm has a certain role in solving the problems with large search space and estimating solutions with high cost [11]. By close look at the process of evolution, i.e. the process applied by nature to solve its problems, it is possible to reach interesting and applicable achievements.

B. Genetic algorithms process

The steps of genetic algorithm are as follows:

1) Generating the initial population: N chromosomes are randomly are created to generate the initial population P(0). The maximum generation (T) and generation counter is set as zero (t = 0), i.e. the solutions to the problem are generated.
2) Population assessment: the fitness value of each chromosome in the population P (t) is calculated by the fitness function.
3) Reproduction or selection: the operator is applied on the population P (t). (Selection of new children and putting them in a new population)
4) crossover: the new children are formed by crossing over their parents.
5) Mutation: It is applied on the population so that the new population P (t + 1) is produced for the generation t + 1.
6) Final test: when one of the conditions of closing, such as t = T is satisfied, the algorithm ends and a sample with the highest value is provided as the optimal solution; otherwise t = t + 1 and return to the assessment step.

C. K-means algorithm

This algorithm is one of the main data mining algorithms presented by Mc Queen in 1976 year[19]. In this algorithm the data are clustered based on k features mentioned for them. All members within each cluster have the highest similarity with each other and the maximum difference members of other clusters. In this algorithm the number of clusters is already specified and they are used in large data sets and when the number of features is a lot.

D. K-means algorithm process

The steps of K-means algorithm are as follows:

1) First K points are elected as the cluster centers.
2) Data sample is assigned to the cluster the center of which has a minimum distance to each data.
3) After assigning all data to one of the clusters, a new point is calculated as center.
4) The second and third steps are repeated until there is no change in cluster centers [14, 19].
5)

E. The proposed algorithm

The use of evolutionary algorithms in data mining algorithms solves many problems of these algorithms. The algorithm is a combination of genetic and K-means algorithms. The combination of these two algorithms solves the premature convergence of K-means algorithm, resolves the local optimum and obtains global optimum. The goal is run all matrix elements in different sizes that represent the entire program (matrix n * n) with minimum power. In this algorithm tasks are assigned to the processors randomly. Then the crowd system was created by these randomly selected solutions and the best solutions are chosen. So there is one objective function that defines the quality of power consumption and thus a number of children are selected to create a new generation and using the cross over, reproduction and mutation operator a new generation is produced. This process is repeated until the best solution (chromosome) is provided in a path. The aim of the proposed algorithm is to minimize power consumption in setting the matrix elements by a nested loop and quality indicator for each chromosome is the completion or lower power consumption to set matrix elements which is determined by chromosome. To reduce the power consumption, it is necessary to assign inter-connected tasks to a processor.

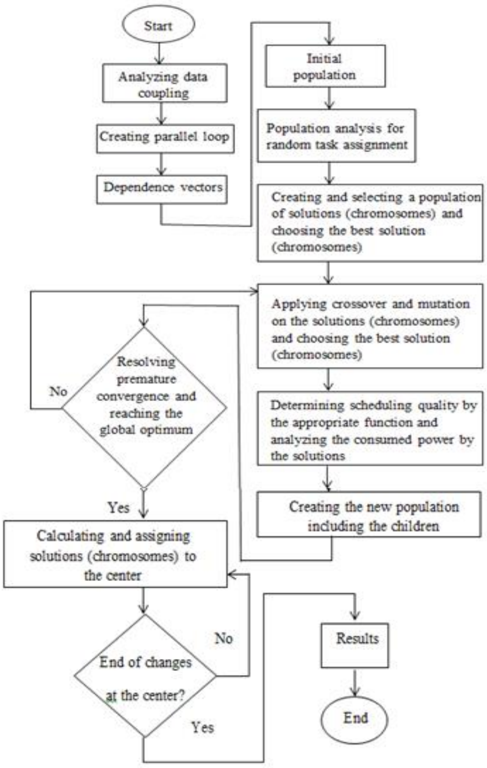
Start
Analyzing data coupling
Creating parallel loop
Dependence vectors
Initial population
Population analysis for random task assignment
Creating and selecting a population of solutions (chromosomes) and choosing the best solution (chromosomes)
Applying crossover and mutation on the solutions (chromosomes) and choosing the best solution (chromosomes)
Determining scheduling quality by the appropriate function and analyzing the consumed power by the solutions
Creating the new population including the children
Resolving premature convergence and reaching the global optimum
No
Yes
Calculating and assigning solutions (chromosomes) to the center
End of changes at the center?
No
Yes
Results
End

Fig. 1: Flowchart of the proposed algorithm

F. SL method

Supervised learning is one of the general methods in machine learning. There is a supervisor in this method that knows the best thing to do in any case. In this way, the system tries to transfer a function from the input to output. Supervised learning requires a number of input data for system training and there is a supervisor that provides data to the learner in addition to the training data [16].

VI. IMPLEMENTATION AND THE TESTED SYSTEM

Implementation is done in NetBeans using the Java programming language. The program used in this study evaluates the power consumption in different matrix sizes by loop interchange technique. Test evaluation was carried out on a system by hardware specifications including: 2.67 GHz processor, 4 GB RAM, 500 GB hard drive and Windows 7 operating systems

VII. RESULTS

Table 1 and Figure 2 present the level of power consumption by the proposed algorithm and supervised learning method at different times and the matrix sizes. The horizontal axis presents the matrix sizes or the number of matrix spaces and the vertical axis indicates the response time (in milliseconds); in this figure the proposed algorithm is prior to the supervised learning method in smaller sizes but in the larger sizes proposed algorithm has better results. According to Table 1 in smaller sizes the response time is lower than the larger sizes and thus less power is consumed.

TABLE I: consumed power in response times (ms) for the proposed algorithm and supervised learning method based on matrix sizes.

Matrix size	Proposed algorithm	Supervised learning [16]
10*20	800000	6000000
10*30	2150000	12000000
10*40	4738000	20000000
10*50	6450000	30000000
10*60	8600000	40000000
10*70	10540000	48000000
10*80	13650000	57000000
10*90	16412000	65000000
10*100	19327000	75000000

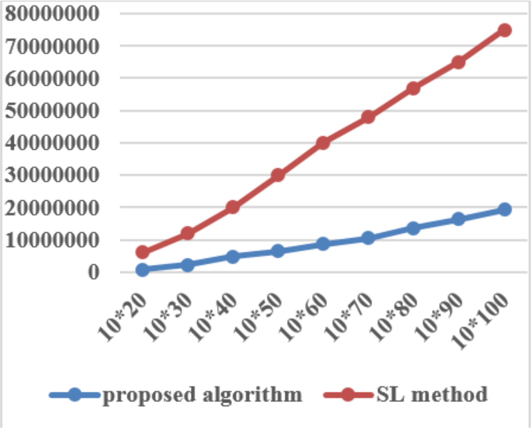

Fig 2: the consumed power for the proposed algorithm and supervised learning method in different response sizes / SL method/ The proposed algorithm

CONCLUSION

In this paper all matrix elements in different sizes representing the entire program segment and the dependency between code segments were implemented on the parallel processors. The aim of this study was to run the entire matrix in different sizes of program segment by minimum power consumption. Accordingly the loop interchange and combination of genetic and K-means algorithms was used. Since this is a N-P hard problem, using a combination of evolution and clustering algorithms the problem was solved by the proposed algorithm that the results indicate that power consumption is improved.

REFERENCES

[1] M. Kandemir, SW So, and G. Chen, "An Evaluation of Code and Data Optimizations in the Context of Disk Power Reduction" Procceding of the International Symposium on Low-Power Electronics and Design, PP: 209-214, San Diego, California, USA, 8-10 Aug, 2005.

[2] A. Oscar, J. Manuel, "A Survey of Software Optimization Techniques for Low-Power Consumption" Proceedings Computing Research Conference, PP: 43 - 46, Estados Unidos, December, 2002.

[3] V, Sarkar, "Challenges in Code Optimization of Parallel Programs" 18 International Conference on Theory and Practice of Software, PP: 1, Springer, Berlin, March, 2009.

[4] V, Sarkar, " Code Optimization of Parallel Programs: Evolutionary vs. Revolutionary Approaches "6 th Annual ACM International Symposium on Code Generation and Optimization, PP: 1, ACM, New York, USA, April, 2008.

[5] D. Saougkos. G. Manis, "Self adaptive run time scheduling for the automatic parallelization of loops with the C2µTC / SL compiler" Parallel Computing, Vol. 39, No. 10, PP: 603-614, October, 2013.

[6] C. Hsu, U. Kremer , "The design, implementation, and evaluation of a compiler algorithm for CPU energy reduction", in Proc. of the ACM SIGPLAN Conf. on Programming Language Design and Implementation, San Diego, California, USA, PP: 38-48, June, 2003.

[7] R. Leupers. "Code Optimization Techniques for Embedded Processors: Methods, Algorithms, and Tools", Kluwer Academic Publishers Norwell, MA, USA, 2000.

[8] N. Edward Johnson, " Code Size Optimization for Embedded Processors" PhD. thesis, University of Cambridge, May, 2004.

[9] PA Kulkarni, DB Whalley , GS Tyson, and JW Davidson, "Practical exhaustive optimization phase order exploration and evaluation "TACO, Vol. 6, No. 1, PP: 1-32, March, 2009.

[10] M. Mitchell, "an Introduction to Genetic Algorithms", MIT Press, Fifth printing, 1999.

[11] DE Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison -Wesley Publishing Company, 1st Edition, 1989.

[12] DA Coley, "an Introduction to Genetic Algorithms for Scientists and Engineers", World Scientific Publishing Company, River Edge, NJ, USA, 1998.

[13] G. Tzortzis, A. Likas , "The Global Kernel k-Means Clustering Algorithm" International Joint Conference on Neural Networks (IJCNN), PP: 1978-1985, July, 2008.

[14] R. Cordeiro de Amorim, "A survey on feature weighting based K-Means algorithms" Journal of Classification (Springer), Vol. 33, No.2, PP: 210-242, July, 2016.

[15] S. Kaxiras and M. Martonosi . "Computer Architecture Techniques for Power-Efficiency ", Morgan and Claypool Publishers, 2008.

[16] D. Saougkos, G. Manis . "Self-adaptive run time scheduling for the automatic parallelization of loops With the C2lTC / SL compiler" Parallel Computing, Vol 39, PP: 603-614, 2013.

[17] K. Cooper, T. Harvey , and T. Waterman, "An Adaptive Strategy for Inline Substitution" In Compiler Construction, ser. Lecture Notes in Computer Science, L. Hendren , Ed. Springer Berlin Heidelberg, Vol. 4959, PP: 69-84, April, 2008.

[18] T. Bostoen, S. Mullender , Y. Berbers, "Power reduction techniques for data center storage systems" ACM Computing Surveys (CSUR), Vol. 45, No. 3, June, 2013.

[19] J. Mac Queen, " Some Methods for classification and Analysis of Multivariate Observations" Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, Vol. 1, PP: 281-297, 1967 .

[20] R. Leupers, P. Marwedel , "Function inlining under code size constraints for embedded processors" In ICCAD '99: Proceedings of the IEEE / ACM international conference on Computer aided design, Piscataway, IEEE Press, and PP: 253-256, California, USA, and Nov, 1999.

[21] K. Cooper, P. Schielke , D. Subramanian, "Optimizing for reduced code space using genetic algorithms" In Proceedings of the ACM SIGPLAN workshop on Languages compilers and tools for embedded systems, PP: 1-9, New York, USA, July, 1999.

[22] DA Ortiz, NG Santiago , "Impact of Source Code Optimization on Power Consumption of Embedded Systems" 6th IEEE International Northeast Workshop on Circuits and Systems and TAISA Conference, PP: 133 - 1 36, Montreal, QC, June, 2008.