Machine Learning support Energy Aware Efficient Multiprocessors Mapping of Real Time tasks

Shahira M. Habashy
Communication, Electronic, and Computer Department, Engineering Collage, Helwan University, Cairo, Egypt

Abstract
Using processor which supported a Dynamic Voltage Scaling (DVS), can lower power consumption by scaling down the processor' frequency while all task sets still don't miss their deadline. Recent usefulness in machine learning techniques inclusive artificial neural networks has led to the evolution of robust and feasible, prediction models for a variety of fields. This paper immediate a unique model which achieve maximum CPU throughput scheduling while using machine learning to predict the performance of all the available CPU at the scheduling quantum granularity. We show how lightweight ANNs can equip highly true performance predictions for a different set of applications thereby aiding to improve scheduling effectiveness. The proposed structure model makes a decision for assigning the current task to a specific processor with determined speed. This decision is based on a number of criteria including the current task period and deadline, and the available processors' current utilization. The proposed model is composed of a number of artificial neurons layers that are trained to promote relationships between the input parameters and the produced aim output target. An ANN was learned to achieve an accurate prediction of a task to processor mapping and that processor suitable operating frequency. Accomplished results were compared to conventional schedulers which demonstrate, significant performance benefits. The proposed model achieved good feasibility performance and minimum power consumption. Comparing our ANN model throughput to other conventional schedulers shows comparable throughput with round robin schedulers.

Keywords: *Multiprocessors, Neural Network, Task Scheduling, Energy Aware.*

1. Introduction

As multiprocessors become more common, they are used in a wider kind of applications, contain real-time and embedded systems. While there are many benefits to use multiprocessor systems, scheduling with these systems can be very complex. Algorithms that are assumed to execute very well on uniprocessor systems, such as Earliest Deadline First (EDF) do not execute as well on multiprocessors [1].The multiprocessor scheduling problem is commonly stated this way: granted a multiprocessor computing system and numerous tasks to accomplish, how does one effectively schedule the tasks to achieve optimal use of the computing resources? In common, a deterministic examine of the solution space to recognize an optimal solution to this NP-complete proposition is computationally and secularly exhaustive. The proposition difficulty depends principally upon the following agents: the number of tasks, execution time of each task, priority of the tasks, the number of processors and their consistency and performance criteria [2, 3]. Present Systems based on multiprocessors immolate high processing capabilities. The energy efficiency of these systems appears primarily from parallel processing confederated with adaptive frequency. Many functions that were traditionally hardwired can now be accomplished in software, causing flexibility and scanty time-to-market for applications deployment. For multiprocessors, the principal power management technique is introduced that minimize energy consumption. Dynamic Voltage Frequency Scaling (DVFS) diminish both the clock frequency and the supply voltage until the application real-time restraint is somewhat exceeded. Those result in reducing the total operating system dynamic power [4, 5]. The processor energy consumption can be diminished by adapting the clock frequency and the supply voltage. The diminution of processor load grants opportunities to the operating system to lower the clock frequency and hence the supply voltage. In real-time systems, the temporality distinction between the end of a task and its deadline, called slack-time, affords clock frequency pliancy. The application is considered to complete before a deadline that can be executed with the minimum frequency [4, 6]. Classic solutions to the proposition of task scheduling and mapping on a multiprocessor system [7, 8, 9] employment an integration of search techniques and heuristics. However these techniques often yield sufficient solutions, the resulting schedule is generally suboptimal. These techniques are also reviewed as they characterized by insufficient scalability and performance.

Artificial neural networks (ANNs) especially are founded to be utilized in a broad variety of fields due to they are expanded to engage in learning relationships between input data and numeral or declarative outputs. The relationships are often difficult to recognize or establish an algorithm for manually, but by using ANNs, good prediction accuracy can be achieved. ANN supported predictors have been shown to be beneficial in precisely predicting target categories for a wider kind of fields such as predicting network traffic and store market prices. Moreover, feedforward ANN such as that usage in this work have minor computation and can be gracefully accelerated using particularize special hardware [10].

In this paper, an energy efficient strategy using DVFS is considered. Our proposed model uses ML to predict the performance of the available processors at the scheduling time granularity. We exhibit how ANNs can afford highly precise performance predictions for a distinct number of applications thereby aiding to upgrade scheduling efficiency. We find that online training is effective in growing prediction accuracy. Notably, our proposed achieves comparable throughput with other conventional schedulers (round robin schedulers).

In the remainder of this paper, related work firstly discusses in Section II, an overview of machine learning technique introduces in Section III. We present the energy and power management model in Section IV, and then explains our implementation in Section V. Section VI distinct our experimentation then finally Section VII discusses the conclusion.

2. RELATED WORK

There have been very few preceding studies convoy on stratifying machine learning to CPU scheduling. Much of those works used a machine learning for scheduling has been to categorize applications, recognize process characteristic and realize the algorithm's execution history. In [11] the authors introduced SLAQ, as a quality-driven scheduling system proposed for large-scale Machine Learning and training jobs while using shared clusters. SLAQ uses as an input an application-specific information beside holds the iterative characteristic of ML algorithms. This led to an increase in the quality of the result models which gained by a large group of ML training jobs. As a result, SLAQ progresses the overall ML system performance, especially under resource contention. The work in [12] by Atul Negi devote ML techniques to study the CPU time-slice utilization manner of some known program algorithms in a Linux system. Learning is executed by an analysis of fixed static and dynamic characteristics of the processes during running. The goal was to detect the most essential static and dynamic characteristic of the processes that can relieve best in the prediction of the processor burst times which decrease the TaT (Turn-around-Time) process. The authors in [10] demonstrated the newness and utility of using a machine and deep learning techniques in designing Heterogeneous parallel processing scheduling and mapping system. Finally, this study has confirmed that applying ANN/DNN predictors for mapping improves the performance over conventional heterogeneous CPU schedulers and memory intent applications by more than 30% with overheads least than (2%). In [1] they examine and try to solve scheduling periodic tasks problem on homogenous multiprocessors by incorporating machine learning techniques. They take equal periods deadlines as an assumption and exhibit an online scheduling policy. Those policies were used to propose scheduling algorithms, diminish the number of preemptions and lower the multiprocessor scheduling algorithm overheads while online. Literature for multiprocessor systems proposed the problem of energy-efficient real-time task scheduling and introduced a number of efficient solving techniques. Authors In [2] applied a genetic algorithm for disbanding the multiprocessor scheduling problem. They implement and design a genetic algorithm which tends to minimize the task graph executed schedule length on a multiprocessor system. The resulting algorithm achieved scalable and adaptable implementation. They introduced several refinements over state-of-the-art approximation which lead to a robust solution. In [13] authors give a novel genetic algorithm used to solve the parallel processor task scheduling problem. The intended algorithm can ensure that every practicable feasible schedule is easy to find with some probability. Subtasks partition Method guarantees that list heuristics can be installed into the algorithm to achieve improvements in the performance. Simulation results show better improvement in solution quality while reducing in computational complexity. In [4] they design a framework to propose an energy efficient parallel processing systems. They used a combination of two techniques to achieve efficient energy. Those techniques are Dynamic Frequency and Voltage

Scaling (DVFS) model and Dynamic Power Management (DPM) model. The framework signal processing applications are supported on Synchronous Dataflow (SDF) modeling. Discovering the application parallelism was performed using a transformation to a single rate form. A machine-controlled scheduling is then executed, diminished the constraint of energy efficiency and applying DVFS and DPM techniques. This architecture model while solving the problem, takes into considering the number of available cores, the per-processor load and the total energy per cycle, traced from time and power calculations of standard modeled applications. [14] Apples energy-efficient for real-time tasks taking into consideration shared resources. They designed two algorithms aimed to minimize energy consumption as soon as all task meet their deadlines. The first proposed algorithm is called STSST. It combines RM scheduling policy with the SRP resources protocol to achieve task synchronization. The second algorithm (DTSST) is introduced as an extension of the STSST algorithm. It is able to reform the slack time created from the early task. The DVS technique is used mainly to adjust the slack time by decreasing the processor speed while the DPM technique can reduce the energy consumption by putting the processor into dormant mode. The experimental results display that the DTSST technique success in reducing energy consumption up to 20.57% ~ 67.66% above the STSST technique and at the same it consumes 43.79% ~ 67.66% lower energy than DS algorithm. In [15] the authors adopted a restricted migration in which a number of different jobs belong to the same task can be distributed on the available processors. Each task executed at different power modes but migration is not allowed after the job execution starting. Maximizing the whole system performance required take into considering the optional parts execution. The final results show the benefits of decreasing the processor speed and changing the operating frequency. The experimental results also show good performance compared to traditional scheduling algorithms for ordering and distributing the jobs among the processors. In [16] scheduling task set on homogeneous multiprocessor platforms was considered using DVFS. A task to processor dynamic priority is assigned taking into consideration the appropriate processor load and each task deadline. DVFS technique was used basically to decrease energy consumption at the core level. The main challenge proposed in this paper was the trade-off between the system achieved performance and its energy consumption. Feasibility, energy and feasibility/energy were used to measure the proposed algorithm performance. Finally, experimental results, show that the proposed technique achieved efficient energy consumption and acceptable performance comparing to other traditional scheduling techniques.

3. Machine Learning Technique

Recently Artificial Neural Networks (ANNs), are continuing to be utilized in an extended difference kind of fields due to their big engage in determining relationships among input data and both numeral or declarative outputs. Those relationships well-informed by the ANNs are often difficult to recognize and program manually but can afford prime prediction accuracies. ANN start by a number of input parameters (estimated to be the first layer) united to the first hidden layer consists of artificial neurons which are then united to other hidden layers. Finally the last hidden layer is connected to one or more output neurons. There is a numerical weight that is assigned to each neuron in the hidden layers. Each weight is multiplied with its corresponding input item. The result of the hidden layer neuron's incoming connections is all added together. The outcome is directed to an activation function (can be a rectified linear, logsigmoid, or similar). The resulting output of these hidden neurons is then directed to the next layer of neurons which may be the output neuron(s). In the end, the output neuron(s) result is the ANN prediction related to the problem solved. Having an ANN accurate predictions required adjusting all the hidden layer weights using a perfect learning method and good training data choosing. Perfect learning is executed with the aid of learning algorithm such as backpropagation that rectifies the weights trying to reach an optimal minima. This lead to lower the prediction error taking into consideration a number of factors such as the error function used, both the estimated and target output. Trying to learn ANN more complex relationships required a design contain more hidden layers and hidden units, however, estate too deep of an ANN can result in overfitting. Overfitting is achieved when the ANN model results near zero error for training data but gives a high error with inexperienced data. Popularize, the ANN to give consistently prediction with the same accuracies for experienced (i.e., training) and inexperienced (i.e., testing) data is a great goal in ANN design. To achieve that goal, many distinct kinds of approaches are applied. One such general custom in machine learning is to disunite the experience data into three deterministic sets in order to not over fit the designed model. The first one is training (70%) set, the second one is the validation (15%) set, and the last one is testing (15%) set. Evaluating how a designed model executes on the validation and test sets permit to moderate the ANN to disapprove its generalization when predicting for inexperienced data [10].

4. Energy and Power Management Models

Energy management is considered an essential factor in real-time embedded systems. Dynamic voltage scaling (DVS) is one of the important algorithms which can powerfully minimize energy consumption [14, 17]. Recently processors support multiple frequencies and voltage levels which can effect on monitoring the energy consumed. However, changing the operating frequency alternates the final task execution time. Achieve reducing in energy consumption may result in a waiver of optimum scheduling [15, 18]

The power consumption pattern P_i for a task T_i is depended mostly on its running frequency as

$$P_i(f)= (\beta_1 f^{\alpha}+\beta_2) \quad (1)$$

Where f is the processing frequency. Estimating the power consumption function constant values, from [19], as $\alpha = 3.94565$, $\beta_1 = 3.89462 \times 10^{-26}$ and $\beta_2 = 0.8453 \times 10^{-9}$. Tasks can show distinct power consumption characteristics by running on different frequency [20]. Each core runs the task in the active state, wasting power which decided by of the current task characteristics and processing core frequency. When a core does not run any task, it stays in a very low-power (idle) case. The total energy consumption for running the whole tasks is obtained by summing power consumption for all tasks (eq. 2). However, a multi-core system that supports DVFS involved the existence of an efficient energy consumption model [20].

$$P = \sum_{i=1}^{N}\sum_{k=1}^{M}\sum_{l=1}^{v_k} x_{ikl}e_{ikl} \quad (2)$$

Where e_{ijk} shows the power consumed by task τ_i when executed on processor p_j at a voltage level v_k, and x_{ijk} is a variable defined as:

$$X_{ikl} = \begin{cases} 1 & if\ \tau_i\ is\ allocatted\ to\ p\ \ at\ voltage\ level\ l \\ 0 & otherwise \end{cases} \quad (3)$$

At that point, the problem of energy-aware task scheduling and mapping can be realized as a power consuming minimized problem in which we want to Minimize

$$\sum_{i=1}^{N}\sum_{k=1}^{M}\sum_{l=1}^{v_k} x_{ikl}e_{ikl}$$

Subject to

1- $F\tau_i \leq di$ for each i, $1 \leq I \leq N$ (4)

Where $F\tau_i$ means task τ_i finish execution time, and d_i means task τ_i deadline.

2- $\sum_{i=1}^{N}\sum_{k=1}^{M}\sum_{l=1}^{v_k} x_{ikl} = 1$ for each i, $1 \leq i \leq N$ (5)

The first constraint (eq. 4) appoints that a task t_i will not miss its deadline. The second constraint (eq. 5) appoints that each task must be operated on only one processor at one voltage level.

4.1. Frequency optimization process

Given a set of homogeneous processors or cores $\{p_1, p_2, \ldots, p_m\}$. For example, NVIDIA's Tegra 2 processor. Each core can be worked at dissimilar power modes. Usually, there are two to eight distinct options ranging from sleep mode (minimum power demand) to maximum frequency operation. The current power modes for each processor are related to their operating frequency $F = \{f_0, f_1, \ldots, f_n\}$, where F shows the number of frequency levels. Operating the core in a Lower power-demand mode achieve a benefit contribute in reserving system energy and also extended the batteries' life. Our paper goal is to search for an optimal schedule that maximizing the reward from the enforcement of optional parts with as low power as possible while meeting all perceptive deadlines. It is assumed that the task to processor allocation is assigned at a task level. No task migration is allowed, a task begins and ends its execution in the same core. Power mode is also elected on a per task basis.

5. Energy Aware Neural Network Scheduling and Mapping Model (EANNSM)

Multiprocessor real-time scheduling intent to implement a task set belong to an application, using the given multiprocessor system, such that each task doesn't exceed its deadline. An application (or task set τ) is supposed to contain a set of n tasks ($\tau_1 \ldots \tau_n$) [21]. Each task τ_i is described by its referring deadline Di, the worst-case execution time Ci, and its period T_i. The task utilization u_i , is given by C_i / T_i. The task set utilization sum U_{Sum} is the sum of the individual utilization of all tasks.

Our ANN model consists mainly of two stages. Figure 1, shows the structure of our model. The first stage consists of two parts (Task Scheduling, and statistics process). The first part (Task Scheduling) determined the next task to be scheduled. In that part, the deadline algorithm was used to determine the next scheduled task. If several tasks have the same deadline and all of them are ready, the task with the earliest arrival time will be chosen first. That part transfer to the second stage the next chosen task and its characteristics (start time, period, and deadline). The second part (statistics process) was designed to collect statistics information from the PEs related to their actual current utilization. The second stage is an ANN-based performance predictor that uses the estimated performance of all PEs and the chosen task characteristic to identify an optimal mapping aiming to maximize system throughput. In that stage, the chosen task is assigned to PE and its starting times are determined. Also, that processor speed is determined. This model tries to hold a major potential for improving workload in the second optimization stage. The model was trained to set all tasks to their less possible workload and configured the PEs to the lowest suggested supply voltage. A scheduling and mapping model aimed at maximizing the path delay slacks based on the training stage. The goal of this model is to distributed tasks on different PEs and let minimum tasks constrained by tighter deadlines so that we can trace looser constraints and at the same time obtain an optimal performance.

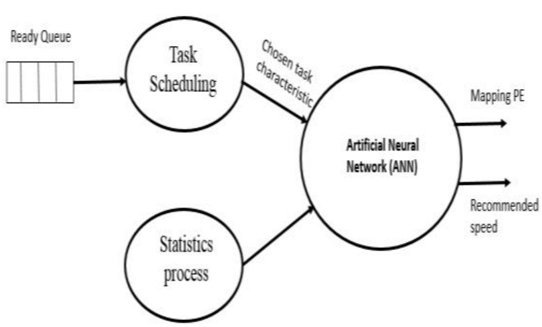


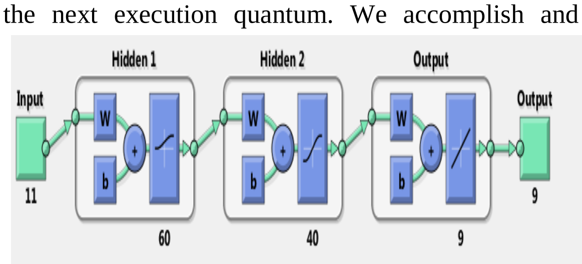
Fig. 1, Energy Aware Neural Network Scheduling and Mapping Model (EANNSM)

5.1. Statistics Estimating Parameter

The number of statistics chosen based mainly on the complexity of the design model and the scheduler optimization scheme. In addition to predicting the performance of a PE based on the next scheduled task characterizing as well as estimating its impacts on the available hardware resources. It is therefore powerful to select a sufficient set of PE statistics to be given as input parameters to our Neural Network predictors. However, that statistical information should be in the generic form. Different statistical values can be applied as inputs to the ANNs. In locating the optimum last set of statistics to be chosen, we search to balance appropriate ANN predictor accuracy as well as trying to decrease the aggregation and arithmetic overheads. After examining how distinct statistics influence ANN prediction accuracies we settled upon the eleven values. The first three values are related to the chosen next scheduled task parameters. Those parameters include that task information (start time, period, and deadline). The next eight values are related to PEs states. Those determine the time each processor from the given eight PEs can be started executing that task. Those eleven parameters will be taken into consideration in determining the best PE to execute that task and that PE execution speed.

5.2. ANN performance predictors

One of the basic contributions of this paper concentrate on the implementation and usage of ANN-based performance predictors. We design a performance predictor, which take as input a set of statistics collected from the chosen task information, and the current PEs parameters. The model outputs are a task to processor mapping decision and that chosen processor speed that the PE is estimated to run during the next execution quantum. We accomplish and



evaluate an ANN model which is given below. The ANN is implemented using the Matlab program.

Fig. 2, ANN used by online performance predictor.

An ANN (shown in Figure 2) formed by 11 input parameters, 2 hidden layers which composed of 60 and 40 hidden units respectively, and 9 output unit. All input units are interrelated to every hidden unit in the

IJCSI International Journal of Computer Science Issues, Volume 16, Issue 3, May 2019
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org https://doi.org/10.5281/zenodo.3252967

first hidden layer. The hidden unit's energizing functions are the log-sigmoid (first hidden layer) and tan-sigmoid transfer function (second hidden layer) while the output unit applied a pure linear transfer function. The designed ANN model aimed to predict a select PE and it's foretell speed. To realize this, we train the predictor with data composed randomly from several executions. After that, we evaluate the precision of the predictor using another random composed data. An error optimization function was used to train that ANN model by measuring the Mean Square Error (MSE) (eq. 6) which is promptly used in ML models.

$$MDE = \frac{1}{n}\sum_{i=1}^{n}(y_i - t_i)^2 \qquad (6)$$

Where in our case y is the predicted value, t is the target one, i is the quantum or sample and n is the total number of quantum.

At the terminal of the second stage, a mapping scheme happens where locates particular task to a specific processor.

6. Analysis and Results

Existing schedulers, which uses a priority or round-robin based heuristic to decide the mapping scheme, are not qualified to improve system performance by predicting a mapping scheme. To do so, a scheduler must recognize or at least be skillful to predict what the system action would be for distinct mapping selections and then choose the one that leads to maximum value. The accurate approach and number of behavioral statistics values for an executing core can be used to predicted system action for different mapping choices. This approach could also model or find relationships between these statistics and the core's resulting performance.

In this paper, we design an ANN performance predictors to get better scheduler's mapping. This progress in scheduler's mapping can be improved by comparing ANN output results with other state-of-the-art schedulers, and show that considerable improvements can be obtained by designing a ML model for homogeneous multicore scheduling. Our experimental results were compared to traditional schedulers which show considerable performance improvement. The precise predictions of the ANN permit our scheduler to get an optimal mapping scheme. We have chosen to use the technique in [16] and the Round Robin Scheduler as the baseline to confirm our proposals since they have been shown to afford good performance improvements for single and multi-processor workloads. The Round Robin techniques used for comparing are First-Fit (FF), Worst-Fit (WF), Best-Fit (BF), and Earliest Deadline First (EDF).

We applied four parameters (throughput, feasibility, energy, and feasibility/ energy) to measure precisely the performance of our proposed ANN scheduling and mapping model. We set the Task utilization factor α to be equal 1 which led to no constraint on any task upper bound utilization. The number of processor M used are 8 processor as in [16]. The Total task set utilization U_{tot} changed from M/ 10 which means (light task load condition) to M which means (heavy task load condition). During the test 1000 task sets were created randomly with varying characteristics (task period P_i, deadline d_i, and utilizations u_i). Each task utilization u_i ranged from 0.0001 to 1. A task set is seen to be feasible if there is a schedule sequence for all tasks on that system which achieve executing all tasks without losing any task its deadline.

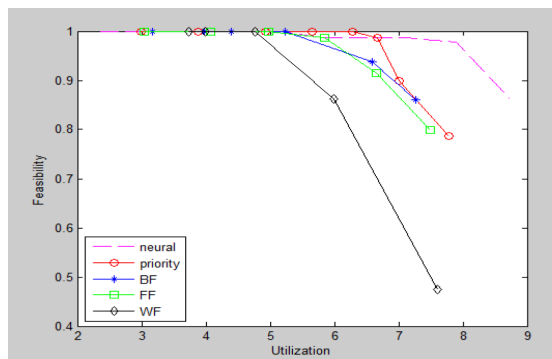


Fig. 3, Feasibility of online ANN performance predictor compared to dynamic priority algorithm, and round-robin schedulers

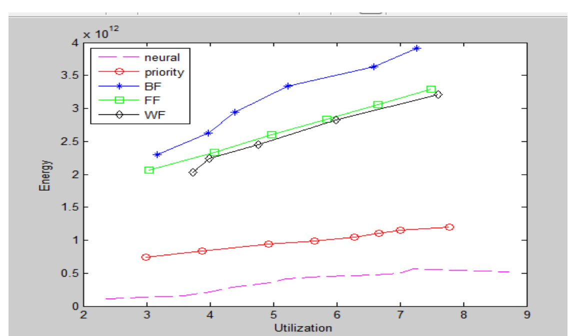


Fig. 4, Energy performance of online ANN performance predictor compared to dynamic priority algorithm, and round-robin schedulers

IJCSI International Journal of Computer Science Issues, Volume 16, Issue 3, May 2019
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org https://doi.org/10.5281/zenodo.3252967

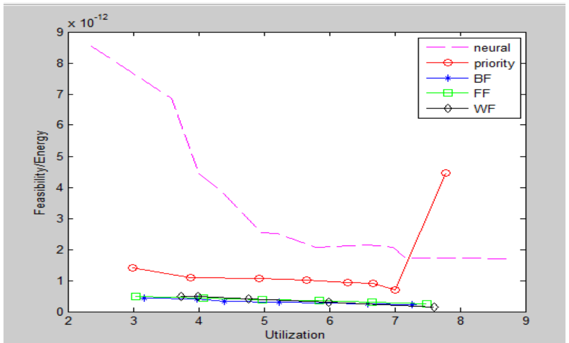


Fig. 5, Feasibility/energy performance of online ANN performance predictor compared to dynamic priority algorithm, and round-robin schedulers

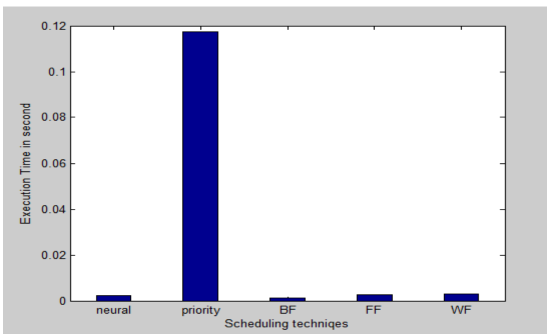


Fig. 6, throughput performance of online ANN performance predictor compared to dynamic priority algorithm, and round-robin schedulers

Figure 3, confirms how ANN can extend highly precise performance prediction for a different set of tasting data thereby achieving progress in improving scheduling efficiency. We show that training can result in the ANN model with perfect prediction accuracy. The figure shows that our ANN model still achieving high feasibility while the system subject to heavy task load condition. Other comparison scheduling techniques (dynamic priority algorithm, and round-robin) couldn't attain that progress.

From Fig. 4, we can realize that the energy consumption of our ANN scheduling model and other differentiation schedulers based mainly on the system utilization. Appling tasks having high utilization will result in increasing the whole system energy consumption. This fact attains in all algorithms because that is when the task execution time increases, the system utilization also increases. In addition, the processor which execute that task must increase its speed in order to achieve task feasibility (don't miss task deadline). Fig. 4, shows that our ANN model predictor get hold of minimum energy consumption compared to other comparison scheduling techniques.

Fig. 5, shows that our ANN model success in progressing very high feasibility/ energy consumption performance compared to other schedulers, especially at light task load. Increasing system task load lead to decrease feasibility/ energy consumption factor but the ANN model still achieves better performance.
In fig. 6, our ANN model affords Convergent throughput performance with comparing to round robin schedulers. On the other hand, dynamic priority algorithm results very high throughput performance comprising to the other exhibited schedulers. That is a result from the increasing number of practicability in dynamic priority algorithm.

7. Conclusion

In this paper, we presented a novel energy-efficient model using artificial neural networks (ANNs) which aiding to estimate system performance leading to approve in the scheduling and mapping sequence. An artificial neural network technique as a predictor is shown to be useful in precisely predicting target grades and values for a wider kind of fields. In this paper, we have to apply an artificial neural network to predict a circuit model representing an optimum energy-efficient multi-core scheduler and mapping system. Our experimental results improved that ANN can provide highly precise performance prediction for a varying kind of task sets. Our approach model provides significant performance improvement compared to other traditional schedulers. Four parameters are applied to measure our model performance and used also for comparison. Those parameters are throughput, feasibility, energy, and feasibility/ energy. The perfect prediction of the ANN permits our schedulers to distinguish an optimal mapping scheme while other convention schedulers cannot. The final results were compared to conventional schedulers (using the same experimental setup) which show considerable performance benefits. Our ANN model can achieve better improvement in feasibility, energy consumption, and feasibility/ energy compared with round-robin schedulers and dynamic priority scheduling techniques. Result average throughput was Convergent comparing to round-robin schedulers while dynamic priority technique was the worst one. Evidently, major performance prediction precision can be realized by concluding more different training data and employing more complex ANNs.

References

[1] Aryabrata Basu, Shelby Funk, "An Optimal Scheme for Multiprocessor Task Scheduling: a Machine Learning Approach", Work-In-Progress Proceedings, 2009.

[2] Michael Bohler, Frank Moore, Yi Pan, "Improved Multiprocessor Task Scheduling Using Genetic Algorithms", Proceedings of the Twelfth International FLAIRS Conference, 1999.

[3] Ya-Shu Chen, Han Chiang Liao, Ting-Hao Tsai, "Online Real-Time Task Scheduling in Heterogeneous Multicore System-on-a-Chip", IEEE Transactions on Parallel and Distributed Systems, Volume: 24, Issue: 1 , Jan. 2013. DOI: 10.1109/TPDS.2012.114

[4] Erwan Nogues, Maxime Pelcat, Daniel Menard, Alexandre Mercat, "Energy Efficient Scheduling of Real Time Signal Processing Applications through Combined DVFS and DPM", 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), Feb 2016, Heraklion, Greece.

[5] Young Geun Kim, Minyong Kim, Sung Woo Chung, "Enhancing Energy Efficiency of Multimedia Applications in Heterogeneous Mobile Multi-Core Processors", IEEE Transactions on Computers, Volume: 66, Issue: 11 , Nov. 2017. DOI: 10.1109/TC.2017.2710317

[6] Adrian Garcia-Garcia, Juan Carlos Saez, Manuel Prieto-Matias, "Contention-Aware Fair Scheduling for Asymmetric Single-ISA Multicore Systems", Computers, Volume: 67, Issue: 12 , Dec. 1 2018.

DOI: 10.1109/TC.2018.2836418

[7] Xiangyu Li, Nijie Xie, Xinyue Tian, "Dynamic Voltage-Frequency and Workload Joint Scaling Power Management for Energy Harvesting Multi-Core WSN Node SoC", Sensors, PMC5336092, 17(2): 310, Feb. 2017.

[8] Yunlian Jiang, Xipeng Shen, Chen Jie, Rahul Tripathi, "Analysis and Approximation of Optimal Co-Scheduling on Chip Multiprocessors", PACT'08, Toronto, Ontario, Canada, October 25–29, 2008.

[9] Ahmed Alsheikhy. "Using Dynamic Moving Average in Real-Time Systems to Minimize Overhead and Response Time for Scheduling Periodic Tasks", IJCSNS International Journal of Computer Science and Network Security, VOL.17 No.4, April 2017.

[10] Daniel A Nemirovsky, Adrian Cristal, Mateo Valero, "Improving Heterogeneous System Efficiency: Architecture, Scheduling, and Machine Learning", phd in Computer Architecture, Department of Computer Architecture Universitat Politècnica de Catalunya, October, 2017.

[11] Haoyu Zhang, Logan Stafman, Andrew or, Michael J. Freedman, "SLAQ: Quality-Driven Scheduling for Distributed Machine Learning", SoCC '17, Santa Clara, CA, USA, September 24–27, 2017.

[12] Atul Negi, "Applying Machine Learning Techniques to improve Linux Process scheduling", IEEE TENCON, Melbourne**,** Qld., Australia, 21-24 Nov. 2005.

[13] YI-WEN ZHONGIZ, JIAN-GANG YANG, "A GENETIC ALGORITHM FOR TASKS SCHEDULING IN PARALLEL MULTIPROCESSOR SYSTEMS", Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi'an, 2-5 November 2003.

[14] Yi-wen Zhang, Cheng Wang, Jin Liu,, "Energy aware fixed priority scheduling for real time sporadic task with task Synchronization", Journal of Systems Architecture, 83 (2018) 12–22.

[15] Isabel M´endez-D´ıaz, Javier Orozco, Rodrigo Santos and Paula Zabala, "Energy-aware scheduling mandatory/optional tasks in multicore real-time systems", INTERNATIONAL TRANSACTIONS INOPERATIONAL RESEARCH, 00 (2016), PP. 1–26

[16] Shahira M. Habashy, "Energy-aware Scheduling based Tasks dynamic Priority on Homogeneous Multiprocessor Platforms", International Journal of Computer Applications (0975 – 8887) Volume 179 – No.16, January 2018

[17] Euiseong Seo , Jinkyu Jeong , Seonyeong Park , Joonwon Lee, "Energy Efficient Scheduling of Real-Time Tasks on Multicore Processors", IEEE Transactions on Parallel and Distributed Systems, Volume: 19, Issue: 11 , Nov. 2008. DOI: 10.1109/TPDS.2008.104

[18] Ajoy K. Datta, Rajesh Patel, "CPU Scheduling for Power/Energy Management on Multicore

Processors Using Cache Miss and Context Switch Data", IEEE Transactions on Parallel and Distributed Systems, Volume: 25, Issue: 5 , May 2014. DOI: 10.1109/TPDS.2013.148

[19] Henri Casanova, et al, "Algorithms and Scheduling Techniques to Manage Resilience and Power Consumption in Distributed Systems", Dagstuhl Reports, Vol. 5, Issue 7, 2016, pp. 1–21.

[20] Abhishek Roy, Hakan Aydin, Dakai Zhu, "Energy-Aware Standby-Sparing on Heterogeneous Multicore Systems", ACM. ISBN 978-1-4503-4927-7/17/06, 2017.

[21] ROBERT I. DAVIS, ALAN BURNS, "A Survey of Hard Real-Time Scheduling for Multiprocessor Systems", ACM Computing Surveys, Vol. 43, No.4, 2011.

Shahira M. Habashy received her B.Sc., M.Sc., Ph.D. degrees in Communications & Electronics Engineering from Helwan University, Cairo, Egypt, in 1997, 2000, and 2006 respectively.
She is currently Associate Professor at Communication, Electronic, and Computer Department, Faculty of Engineering, Helwan University, Cairo, Egypt.