CHEMICAL ENGINEERING TRANSACTIONS
VOL. 88, 2021
A publication of AIDIC
The Italian Association of Chemical Engineering
Online at www.cetjournal.it
Guest Editors: Petar S. Varbanov, Yee Van Fan, Jiří J. Klemeš

ISBN 978-88-95608-86-0; **ISSN** 2283-9216
DOI: 10.3303/CET2188124

Innovative Energy Management System for Mobile Processors Power Consumption with Integration of Predictive Neural Network Models

Elena A. Kirillova*, Alexey I. Lazarev

Department of Information Technology in Economics and Management, branch of the National Research University Moscow Power Engineering Institute" in Smolensk, Energy passage 1, 214013 Smolensk, Russia
no@sbmpei.ru

At the present stage of information technologies development in the field of portable devices implementation one of the perspective direction is the development of software and hardware to optimise the consumption of energy resources in mobile processors. Modern solutions are aimed at continuous improvement of methods to reduce power consumption along with increasing the performance of devices, but the problem of limited time interval of portable devices active use is still relevant. Existing solutions are aimed at using various energy-dependent methods in the kernel configuration – I/O scheduler governors, TCP overload algorithms, as well as additional entropy distribution systems and memory release algorithms. The solutions considered involve a system of aggressive behaviour based on the use of user-driven methods to terminate energy demanding processes in the system. The purpose of this study is to overcome the problems under consideration by developing a cross-platform algorithmic approach, which is based on the tracking of the energy consumption processes in the kernel, taking into account system calls and background activity in the system. A distinctive feature of the proposed solution is the use of a neural network training sample for the processes of tracking user behaviour, which affects the reduction of the CPU load by completing side processes and increasing the time interval of battery performance. To implement the project, root access to the system was also used, assuming full-function access to the system kernel. Another feature of the implemented algorithm is backwards compatibility of the work with mobile processors, which allows to organise work on mobile devices.

1. Introduction

The current development of information technologies is primarily aimed at the market for the production of innovative products in the field of mobile architecture. Manufacturers of software and hardware solutions are currently improving algorithms and methods to improve system performance in parallel with reducing power consumption. This area also affects the electric power and chemical industries due to the automation of individual sub-processes of the operation of the entire system, in particular, when using uninterruptible power supply systems to target installations. Among other things, Rego et al. (2019) describes a methodology for using neuro-fuzzy forecasting of biogas in the chemical industry, and Monteiro et al. (2018) describes the possibility of neural network data processing to detect measurement errors when preparing experimental samples. The considered methods and algorithms, integrated into mobile devices, offer the user a flexible system for ensuring efficient operation over a long period of time, up to the use of artificial intelligence for training and optimising energy consumption algorithms. However, the problem of active use of the device in a short time interval remains relevant.

The solution to this problem from third-party developers is the use of various energy-dependent methods, among which there are governors, I/O schedulers, TCP overload algorithms, entropy distribution systems and volatile memory release. Examples of control system implementations in the Android operating system can be software solutions from the developers of ElementalX, Franco Kernel. Also important is the solution to the problems of centralised CPU load balancing described by Zhao et al. (2020). All the considered indicators are controlled at the kernel level and allow you to significantly reduce the coefficient of active and passive energy

Paper Received: 29 April 2021; Revised: 29 September 2021; Accepted: 10 October 2021
Please cite this article as: Kirillova E.A., Lazarev A.I., 2021, Innovative Energy Management System for Mobile Processors Power Consumption with Integration of Predictive Neural Network Models, Chemical Engineering Transactions, 88, 745-750 DOI:10.3303/CET2188124

hourly consumption, while the user may face obvious problems of insufficient memory allocation, excessive increase or decrease in efficiency due to the choice of an incorrect governor. The publication of Harini and Ravikumar (2020) demonstrates the dependence of dynamic scaling of the voltage and frequency of the CPU in order to increase efficiency through parallel processing. Rothman (2021) describes the main provisions of implementing neuro-fuzzy systems in software solutions. The value prediction model uses as the main process a bidirectional recurrent neural network, described by Nathwani (2020) for solving signature recognition problems, and allows determining the dependence between parameters based on a training sample.

Although the solution of the described problem is currently being paid quite close attention by Russian and foreign researchers, all existing methods have a common drawback, namely, they assume the user completion of energy-demanding processes in the system, which is not always convenient and possible. The algorithms used in the standard representation to reduce CPU and GPU load are characterised by a critical change in the operating mode due to the accurate determination of the values of system parameters in real time, which in turn does not always affect the optimal operation of a particular device. This determined the research objective to design a cross-platform system based on the analysis and management of system calls and background activity of the device without the direct involvement. From a technical point of view an essential feature of the proposed solution is the modular architecture for tracking system load statistics. The system is based on an algorithm that allows using artificial neural networks and fuzzy logic to determine the critical points of increasing the processor frequency, adjusting the kernel parameters, and ensuring stable operation of the processor with minimal energy consumption, which determines the theoretical novelty of the proposed method. The practical significance is determined by the fact of tracking user and system processes with the possibility to reduce energy consumption.

2. Algorithmic implementation of the system

The main requirements highlighted for the algorithm to be implemented are multiplatform and the ability to work with most ARM family processors. A secondary requirement for the implemented algorithm is to have root access to the file system due to the need to handle kernel parameters and system processes, described in the paper on kernel optimisation by Yuan et al. (2018). An alternative important feature of the implemented algorithm is support for working with custom kernels and governors on the Android OS platform, due to the direct dependence of CPU regulators on energy efficiency (Hoffman et al., 2020).

The algorithm for identifying energy-dependent parameters is based on determining the type of available options and then presenting them as factors processed by the neural network. To implement a modular architecture for tracking background processes, a hierarchical algorithm was designed, as illustrated in Figure 1. The presented algorithm is processed by parallel processing of the following modules:

- Module “define available kernel functions” is used to initially identify the system parameters supported by the kernel. These parameters include CPU and GPU controls, stun and boost options, as well as minimum and maximum allowable CPU frequencies;
- Module “track usage statistics” allows you to accumulate information about user activity, the most energy-significant processes and determine the level of CPU load;
- Module “predictive identification” module is used to predict the values of various factors for subsequent decision-making;
- Module “decision-making” optimises the system and then compares the results;
- Module “optimising the operation of the governor” assumes the possibility of changing the system governor to force a decrease or increase in the processor frequency.

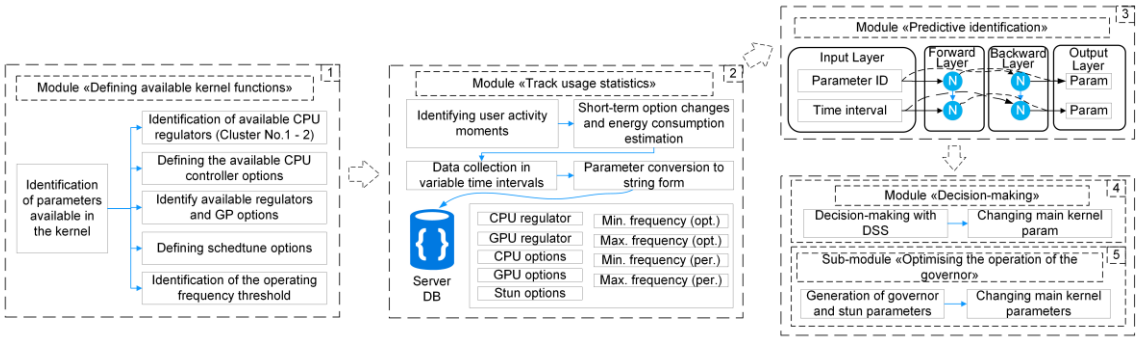

Figure 1: The algorithm of the implemented system for reducing energy consumption

2.1 Statistical data accumulation module

The data accumulation module intelligently determines the moments of user activity. The main activation function of this module is the peak moments of unloading and loading processes and the analysis of the current load level. This module involves converting the parameters to a string form, which are then written to the corresponding database table. The database is represented by a PostgreSQL server deployment based on the construction principles described by Schönig (2020). As can be seen from Figure 2, each parameter group was divided into separate tables with corresponding relationships. The adaptation of the structure under consideration made it possible to record the original and predicted patterns in a single hierarchy.

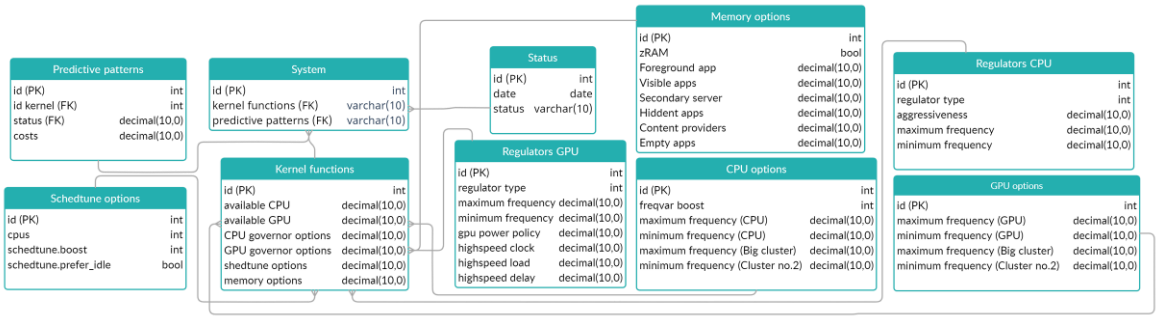

Figure 2: ER-diagram

2.2 Predicting available parameters

The predictive parameter processing module in this software uses the prediction of system calls that have a direct impact on the excessive increase in the discharge rate. As you can see from Figure 3, the non-optimised operation of the kernel at certain points in time causes the discharge rate to increase to 30 %/h. This aspect of the work mainly affects the use of the processor governor, which increases the CPU frequency to the maximum, as well as the dependencies of the memory options table in the structure of the ER diagram.

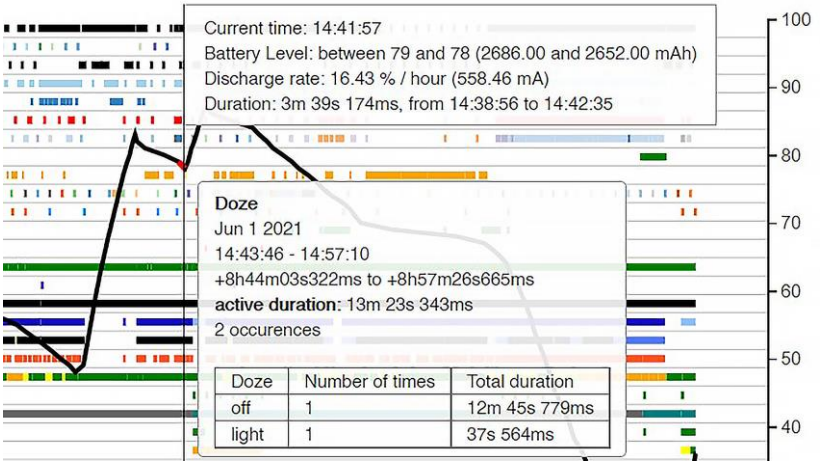

Figure 3: Analysis of abnormal operation of the energy consumption system

According to research by Yan et al. (2019), the methodology for measuring energy consumption is divided into several detailed procedures, analytically described by the combination of consumption of various factors in Eq (1). The E_{smart} indicator describing the power consumption based on the processor is reflected by the Eq (2).

$$E_{total} = E_{smart} + E_{BS} + E_{wireline} + E_{DC}, \quad (1)$$

were E_{total} – total power consumption, W; E_{smart} – CPU-based power consumption, W; E_{BS} – mobile base station power consumption, W; $E_{wireline}$ – mobile wired power consumption, W; E_{DC} – mobile data centre power consumption, W.

$$E_{smart} = (\int_0^{D_{sig}} P_{S\_CPU} + \int_0^{D_{sig}} P_{S\_4G})dt, \quad (2)$$

were E_{smart} – total power consumption, W; D_{sig} – signal duration, s; P_{S_CPU} – processor power consumption, W; P_{S_4G} – 4G module power consumption, W.

Since the parameters under consideration have a dynamic representation of the float type, their prediction is carried out using a bidirectional recurrent neural network (BRNN), as described in the publication Bohan and Yun (2019) of time series forecasting. An essential feature of the application of this neural network is the ability to predict input data in a bidirectional time interval due to the implementation of recurrent directed neurons that work in the opposite direction. Among other things, Long-Short Term Memory (LTSM) belongs to the category of a neural network with a long-term memory type, which allows you to use the dependencies identified in a variable time interval in order to record notable features, described by Goldberg (2017). The standard implementation of LTSM is based on the use of the following equations:

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f), \tag{3}$$

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i), \tag{4}$$

$$m_t = W_{mx}x_t + W_{mh}h_{t-1} + b_c, \tag{5}$$

$$c_t = f_t \bullet c_{t-1} + i_t \bullet m_t, \tag{6}$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o), \tag{7}$$

$$h_t = \tanh(c_t) \bullet o_t, \tag{8}$$

were x_t – input vector; h_t, h_{t-1} – time output vector «t», referring to the value of t-1; i_t – output activation vector; W_{ix}, W_{ih}, b – weights of the representation of vectors for each step.

The software implementation of the considered neural network is based on the use of the machine learning library PyTorch Pointer (2019). The validity of the choice of the PyTorch library is due to the advantage of the software based on the Python language, which allows you to run program code on multi-platform operating systems-based on the Linux kernel, OSX, as well as the Microsoft Windows OS family. The input vector (x_i) processes the input data - the CPU load, as well as other dynamic parameters of the kernel using the hidden states of the forward and reverse inputs, in this connection, the activation vector of the hidden layer takes the form Eq (9 – 10):

$$h_t = W_x^t x_t + W_h^t h_{t-1} + b_h^{t-1}, \tag{9}$$

$$h_t^{'} = W_{x'}^t x_t + W_{h'}^t h_{t+1} + b_{h'}^{t+1}, \tag{10}$$

were W – the weight of the argument in forward and backward propagation; h – the weight of the hidden layer for processing the value in the time intervals t-1, t+1; b – offset vector.

So, the input data for this neural network are floating values, for example, the CPU is represented as “1,267; 1,363; 1,925”, while the output data has an identical form of predicted values. To operate a bidirectional ANN, the layers.Bidirectional function is added with the appropriate parameters, with a function value of zero for the reversed layer. The processing of the layer weight on a single interval with the subsequent use of the reverse gru function is noteworthy. Testing of the implemented software component was carried out using an Intel Core i7-10510U processor, as well as a GeForce GTX 1650 Max-Q design video card compatible with CUDA technology.

As a result of the operation of a single instance of the output neural network, the output vector represents floating values, which are represented by inputs in the fuzzy logic apparatus. Table 1 shows the results of neural network training, according to which the data “accuracy” and “losses” increase and decrease in the process of neural network training, which indicates successful training.

Table 1: Neural network training results

Patterns, n	50	100	200	400	600	800	1000
Loss, %	1,743	1,562	1,424	1,125	0,984	0,812	0,772
Accuracy, %	1,934	1,836	1,743	1,472	1,293	0,946	0,852

2.3 Decision-making based on the fuzzy logic apparatus

The fuzzy logic unit processes the output values of the neural network for appropriate decision – making on the termination of individual processes or changing the operating mode of the governor. Fuzzy rule interpolation is based on the Sugeno inference model, an improvement of which is proposed in the publication of the authors Zhang and Shen (2019). The software implementation of this module is based on the FuzzyWuzzy library, which allows the use of floating value character comparison functions for subsequent decision-making. Processing the output values of the neural network vectors allowed us to obtain the following results when calling the “fuzz.Ratio” function:

```
cpu_state_current = [“806, 650, 1404, 520, 1300, 949, 2002, 520”]
cpu_state_predictive = [“445, 650, 1508, 624, 1742, 949, 1222, 520”]
fuzz.ratio(cpu_state_current, cpu_state_predictive)
IF (DISCARGE_RATE > 10.0) THEN CALL MODULE_STATE;
IF (E_TOTAL > 500) THEN CALL MODULE_ANALYZE;
IF (DISCARGE_RATE > 10.0) AND (TIME > 3) THEN CALL MODULE_EOPTIMAL;
IF (DISCARGE_RATE > 15.0) AND (TIME > 7) THEN CALL MODULE_ESAVING;
```

The version of the rules presented above also allows you to separately use the modules No. 4-5 in Figure 1, that is, it can be used both to change the main parameters of the kernel and to make changes to the operating mode of the governor. Testing of the operability of this set of rules was performed both on a personal computer with Microsoft Windows OS, and using the Android OS operating system with Python modules installed. As a result of the operation of this module, it can be concluded that the joint use of a neural network instance together with the fuzzy logic module allows you to make decisions on adjusting the operation of the device in order to reduce power consumption.

2.4 Analysis of test results

As a result of the development of the software module, the impact of the implemented software was tested on the Samsung Galaxy Note 10 (N970F) device. As can be seen from Figure 4, the integration of the developed algorithm allowed to reduce the process of energy consumption by eliminating the moments of excessive increase in the discharge rate parameter.

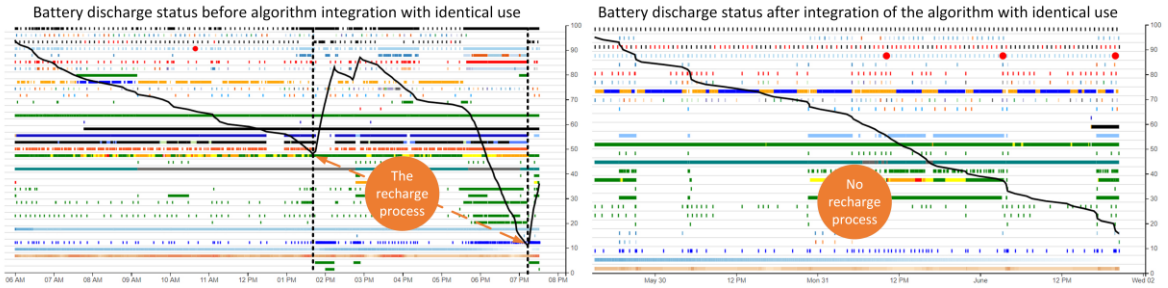

Figure 4: Testing the performance of the implemented software

Figure 4 also shows the user's activity period, which varies from 10 h (the original graph) to several days (the resulting graph). Also important is the process of reducing the recharge cycles of the Li-ion battery, which affects the overall battery life. In the future, it is planned to conduct testing on other types of processors, finalize the proposed software solution for a wider range of consumers and register such a software package.

3. Conclusion

Currently, it is quite acute the problem of insufficient optimization of the active power consumption of mobile processors was highlighted. The methods proposed by Russian and foreign scientists to overcome it have significant drawback related to the need for active user interaction with the hardware and software system. In this regard, the authors proposed the development of a cross-platform system for optimising power consumption due to analyses of dynamic and static factors was carried out. The analysis of active energy consumption showed that the peak values of the energy consumption indicator per hour reach the level of 30 %/h. During the analyses of devices operation also the main energy-dependent factors of the CPU, GPU, as well as application factors that are worked with using the PostgreSQL database were highlighted. The article presents an algorithm that optimizes the power consumption of mobile processors.

The novelty of the proposed approach is a software solution for managing the operation of mobile processors to optimize energy consumption without active user intervention, which was implemented as follows. At the

stage of predicting the available parameters, a bidirectional recurrent neural network was integrated into the implemented system, which made it possible to use the trained patterns to predict the main parameters of the system – active and passive energy consumption, processor frequencies to determine and reduce critical moments of electricity consumption. To reduce energy consumption during operation, a method of neural processing based on previously recorded patterns was introduced. The implementation of the module of the fuzzy logic and decision-making apparatus is based on the use of specialised algorithms and subsequent decision-making based on the conditional comparison rules given in the article. The software implementation of the modules under consideration was carried out using the Python libraries – PyTorch, FuzzyWuzzy described by Vishnu (2018).

Confirmation of the effectiveness of the developed algorithm is reflected in the graphs of the analytical comparison of energy consumption, where the indicator of active energy consumption was reduced to 2%/h, resulting in the conclusion that the algorithm can be implemented in systems with an ARM processor.

The developed approach reveals wide opportunities for the restructuring of energy management systems for mobile processes. A promising increase in the efficiency of ARM systems, along with a decrease in energy consumption, is possible due to the development of a universal behaviour model based on multiple models of daily operation of mobile devices. The development of a universal neural network model will optimise the process of mobile penetration testing, where energy consumption has a significant impact on the environment of the chroot container. The development in the IoT segment will allow us to organise a more efficient system of autonomous object management in enterprises of various sizes.

References

Bohan H., Yun B., 2019. Traffic Flow Prediction Based on BRNN, 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China, 12-14 July, 320-323.

Goldberg Y., 2017. Neural network methods in natural language processing, Morgan & Claypool Publishers, San Rafael, CA, USA.

Harini S., Ravikumar A., 2020. Effect of Parallel Workload on Dynamic Voltage Frequency Scaling for Dark Silicon Ameliorating, International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 10-12 September, 1012-1017.

Hoffman A., Pathania A., Kindt P.H., Chakraborty S., Mitra T, 2020. BrezeFlow: Unified debugger for android CPU power governors and schedulers on edge devices, 57th ACM/IEEE Design Automation Conference (DAC), San Francisco, USA, 5-9 December, 1-6.

Monteiro F., Valim I., De Siqueira R., Moura F., Grillo A., Santos B., 2018. Application of Artificial Neural Networks for Identification of Catalysts Used in Thermogravimetry Lignocellulosic Biomass, Chemical Engineering Transactions, 65, 529-534.

Nathwani C, 2020. Online Signature Verification Using Bidirectional Recurrent Neural Network, 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 13-15 May, 1076-1078.

Pointer I., 2019. Programming pytorch for deep learning: Creating and deploying deep learning applications, O'Reilly Media, Farnham, UK.

Rego A., Leite S., Leite B., Grillo A.V., Santos B.F., 2019. Artificial Neural Network Modelling for Biogas Production in Biodigesters. Chemical Engineering Transactions, 74, 25-30.

Rothman D., 2021. Transformers for natural language processing: build innovative deep neural network architectures for Nlp with Python, PyTorch, TensorFlow, Bert, RoBERTa, and more, Packt Publishing, Birmingham, England, UK.

Schönig H., 2020. Mastering PostgreSQL 13: Build, administer, and maintain database applications efficiently with PostgreSQL 13, 4th ed., Packt Publishing, Birmingham, England, UK.

Vishnu S., 2018. Deep Learning with PyTorch: A practical approach to building neural network models using PyTorch, Packt Publishing, Birmingham, England, UK.

Yan M., Chan C.A., Gygax A.F., Yan J., Campbell L., Nirmalathas A., Leckie C., 2019. Modeling the total energy consumption of mobile network services and applications, Energies, 12 (1), en12010184.

Yuan P., Guo Y., Chen X., Mei H., 2018. Device-Specific Linux Kernel Optimization for Android Smartphones, 6th International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), Bamberg, Germany, 26-29 March, 65-72.

Zhang P., Shen Q., 2019. A Novel Framework of Fuzzy Rule Interpolation for Takagi-Sugeno-Kang Inference Systems, International Conference on Fuzzy Systems, (FUZZ-IEEE), New Orleans, USA, 23-26 June, 1-6.

Zhao S., Hao C.-L., Zhai J., Li M.-S., 2020. S-Bridge: CPU Load Balancing Agent for Performance Asymmetric Multicore Processors. Ruan Jian Xue Bao/Journal of Software, 31 (9), 2965-2979.