Software Service to Optimize Battery Life in a Personal Computer

[1] Nadine V Alexandrine, [2] Kavitha S N

[1] Post Graduate student, Software Engineering, RV College of Engineering,
[2] Assistant Professor, Department of Information Science and Engineering, RV College of Engineering

Abstract: **Many high-end devices in industries or for a naive user, such as laptop computers and mobile phones, use rechargeable batteries that allow a user to recharge the battery once the battery is been drained. Hence, the battery is a component and the end-user will focus on the performance of it. Optimizing the battery without affecting the performance is the focus of this paper. A software service to optimize battery in a laptop is developed focusing on areas such as Information gathering, CPU utilization, Application background colour, Monitor brightness and Foreground Application.**

Index Terms **– Optimization, CPU Utilization, Foreground Application, Windows Management Instrumentation.**

I. Introduction

A battery in a laptop is a part of hardware that supplies a computer with energy to operate, allowing that device to operate without a power cord. Depending on how much power it takes, batteries are often able to power a laptop computer for many hours. High-end machines, such as laptop computers and mobile phones, use batteries that are rechargeable.

Lithium ion batteries are used in laptops, with the flatter lithium polymer technology being used in some thinner versions. The older nickel metal-hydride batteries have largely been replaced by these two technologies. The battery life varies greatly by the model and workload that it is designed for, and can vary from an hour to a single day.

Laptop batteries can also be costly since the only one that makes a battery to suit their individual laptop is the laptop maker. This is done so that when a new battery is needed, the manufacturer can have control of the demand. This is mostly done to motivate, instead of replacing the battery, to purchase a new laptop. In certain cases, the price, which is what the manufacturer intended, makes no difference.

The battery is the one of the focus points for an end user to purchase the machine. The battery is optimized using three factors that is CPU utilization, Monitor brightness and Application background colour currently. The system Optimization is also applied to enable usage of battery in a machine. These three factors can increase the battery usage of a personal computer for a naïve user.

II. Proposed System

The goal is to develop a Service that optimizes Battery usage in a Personal Computer. Optimizing Battery without compromising on Performance is the focus and this paper is specific for Windows Operating System.

A. OBJECTIVES

- Monitor the system for Process level Details.
- Identifying the User's Action, the foreground application.
- Building the Battery Optimizer Engine to detect CPU Utilization, Monitor Brightness and Application Background Color.

The main aim is to develop a service that Optimizes Battery in a personal computer. Monitoring the system for Information is done initially to understand what is the current state of the system and its processes. This information is structured and stored.

Identifying the user's action focuses on capturing the foreground application only, the application that is visible to the user and maximized and the application that is highlighted in case two applications are minimized adjusted according to the screen size.

The Factors from application perspective is to focus on CPU Utilization, Monitor Brightness and Application Background Colour.

III. System Architecture

This section covers the high level architecture, comprehensive flowchart, requirement analysis.

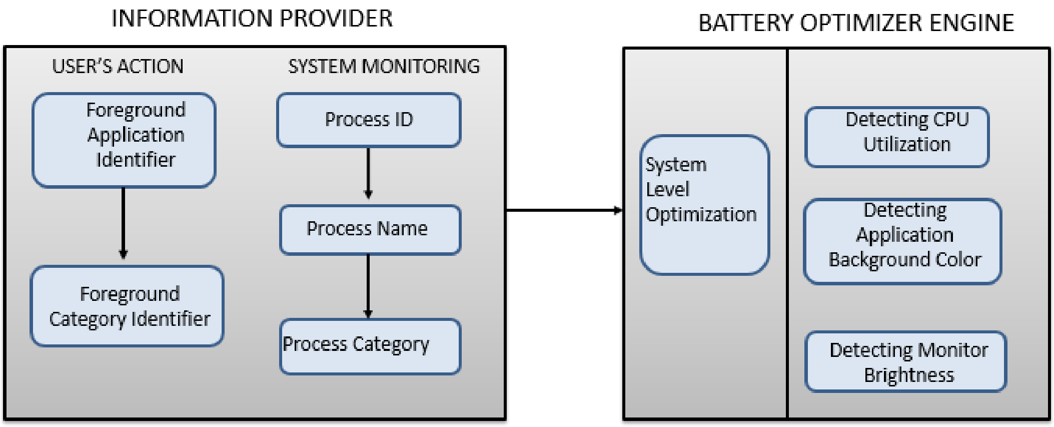
INFORMATION PROVIDER
USER'S ACTION
SYSTEM MONITORING
Foreground Application Identifier
Foreground Category Identifier
Process ID
Process Name
Process Category
BATTERY OPTIMIZER ENGINE
System Level Optimization
Detecting CPU Utilization
Detecting Application Background Color
Detecting Monitor Brightness

Fig 1. System Architecture of the Proposed System

The architecture of the proposed system; the information provider has two divisions, User's Action and the System Monitoring. The system monitoring and the user's action is a single thread. The continuous change of applications is captured in the Foreground Event which is the foreground Event identifier. The process level details are captured and stored continuously every time the thread functions.

System Monitoring captures the process level details in a thread. There is a repository created for analysing and categorizing each process details. The process details are continuously captured and stored with a new set to avoid any garbage value. The system category is roughly divided into ten categories. These categories will later help to analyse the optimization that has to be done against its processes.

User’s Action is an important module which offers the second base to analyse the type of application each system is working on. The thread works continuously to find the foreground application and stores the information of the foreground application. The foreground category identifier would then lookup the repository and capture the category of the foreground process. This is a continuous process and works in a multithreaded manner.

System Level Optimizations are applied depending on the type of OS and the current specifications. Application Level Optimization are detecting CPU Utilization which is a major key role contributor to battery usage. The CPU utilization which is consumption of Processor time with respect to time is considered. Application background color is another factor which can contribute to the battery usage and hence the further actions to decrease Monitor brightness seamlessly was done.

IV. Design

The diagram in Fig 2 describes the workflow of the proposed system. The Event Hook continuously monitors the foreground and the other threads monitor the CPU utilization, Monitor Brightness and Application Background Color.

There are two parts to the implementation, the Information Provider and the Battery Optimizer Engine. The Information Provider is built to improve the battery optimizer engine to decide if the optimization is required or not. The information provider has two divisions, User’s Action and the System Monitoring. The system monitoring and the user’s action is a single thread. The continuous change of applications was captured in the Foreground Event which is the foreground Event identifier. The process level details are captured and stored continuously every time the thread functions.

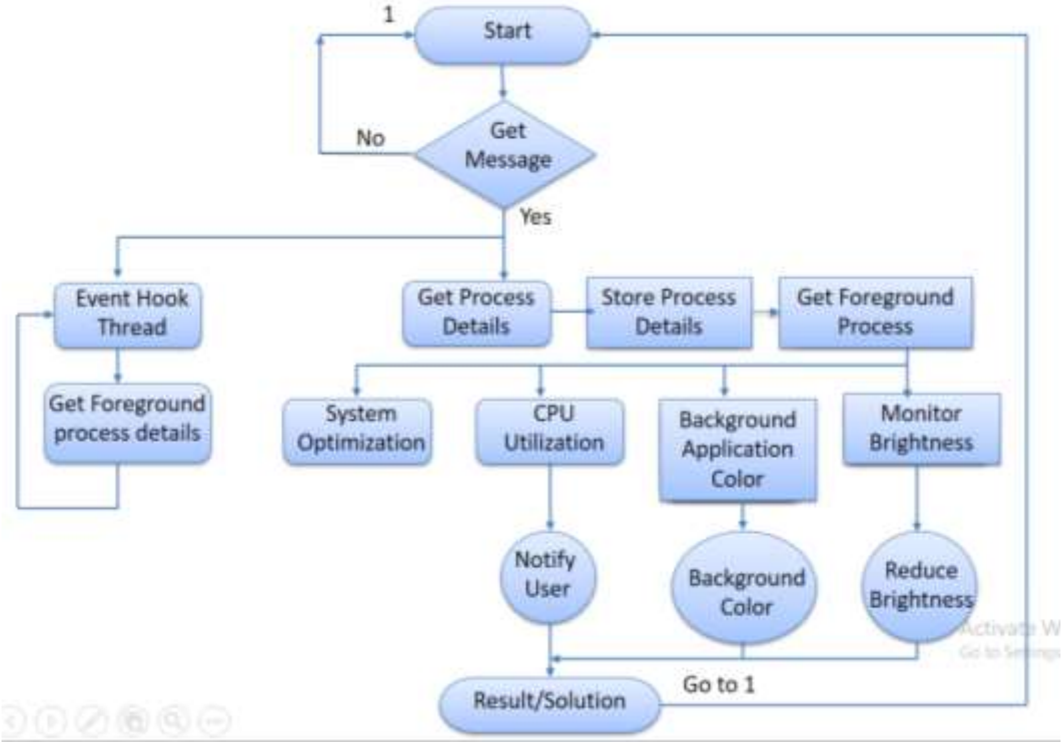

Fig 2. System Architecture of the Proposed System

V. Results

The executable file, can be installed like any other application to monitor and increase Battery Life. The executable file focuses for Windows Operating System. The result is seen for any naïve user or a user with a load of 30% on the system.

Future Developments

The enhancement of the proposed system can be extended to work on the process of once the highest CPU utilization is captured and the user will be notified of it. The controls can be enhanced taking user's choice on whether the user would like to reduce the brightness and choose a lighter background color for the application.

VI. Conclusion

The system currently gathers information of all process level details in a time interval of ten seconds. The information is gathered and stored as a lookup table. This lookup is refreshed when the module is called. The Battery Optimizer engine works on three factors CPU Utilization, Monitor Brightness and Application Background Colour.

Acknowledgement

I would like to extend my sincere thanks to my guide, **Prof. Kavitha S N,** Assistant Professor, Department of Information Science & Engineering, RV COLLEGE OF ENGINEERING, Bengaluru for constantly supporting and providing her valuable feedback.

Special mention to **Mr. Koushik Mondal** and **Mr. Abir Kumar** who supported me with their ideas and thoughts. Their help and support is commendable.

I would also like to express my sincere gratitude to Dr. B. M. Sagar, Head of the Department, Information Science & Engineering, RV COLLEGE OF ENGINEERING, Bengaluru for his suggestions, constant support and encouragement.

I would also like to express my sincere gratitude to Dr G.S Mamatha, Prof & Associate Dean, Information Science & Engineering, RV COLLEGE OF ENGINEERING, Bengaluru for her constantly guiding and motivating through her thoughts and ideas.

I would also like to thank Dr. K. N. Subramanya, Principal, RV COLLEGE OF ENGINEERING, Bengaluru, for his support and guidance.

I thank my Family, and all the Faculty members of Department of Information Science & Engineering for their encourage and support.

I would like to express my deepest appreciation and gratitude to all those who helped me to complete this project.

References

[1]T. Imai and H. Yamaguchi, "Dynamic battery charging voltage optimization for the longer battery runtime and lifespan," *2013 IEEE 2nd Global Conference on Consumer Electronics (GCCE)*, 2013, pp. 219-223, doi: 10.1109/GCCE.2013.6664804.

[2] W. Lin, K. Yarn, T. Cheng, S. Chen and S. Wang, "Lead-Acid Battery Monitoring System by Using PC Graphic Control," *2009 International Conference on Electronic Computer Technology*, 2009, pp. 153-156, doi: 10.1109/ICECT.2009.93.

[3] P. Nuccio, M. Michaud and S. Gentile, "Optimizimg the mobile PC battery through smart battery software," *The Twelfth Annual Battery Conference on Applications and Advances*, 1997, pp. 201-204, doi: 10.1109/BCAA.1997.574103.

[4] B. Dickinson and J. Gill, "Issues and benefits with fast charging industrial batteries," *Fifteenth Annual Battery Conference on Applications and Advances (Cat. No.00TH8490)*, 2000, pp. 223-229, doi: 10.1109/BCAA.2000.838408.

[5] S. Kunii, "Means of realizing long battery life in portable PCs," *1995 IEEE Symposium on Low Power Electronics. Digest of Technical Papers*, 1995, pp. 20-23, doi: 10.1109/LPE.1995.482448.

[6] Ying-Wen Bai and Hsiu-Chen Chen, "Design and Implementation of Low-Power Tablet PCs," *2006 IEEE International Symposium on Consumer Electronics*, 2006, pp. 1-6, doi: 10.1109/ISCE.2006.1689435.

[7] A. Dinc and M. Otkur, "Optimization of Electric Vehicle Battery Size and Reduction Ratio Using Genetic Algorithm," *2020 11th International Conference on Mechanical and Aerospace Engineering (ICMAE)*, 2020, pp. 281-285, doi: 10.1109/ICMAE50897.2020.9178899.

[8] S. Karimi and F. Tahami, "A Comprehensive Time-domain-based Optimization of a High-Frequency LLC-based Li-ion Battery Charger," *2019 10th International Power Electronics, Drive Systems and Technologies Conference (PEDSTC)*, 2019, pp. 415-420, doi: 10.1109/PEDSTC.2019.8697876.

[9] Y. Wang, Y. Li, L. Jiang, Y. Huang and Y. Cao, "PSO-based optimization for constant-current charging pattern for li-ion battery," in *Chinese Journal of Electrical Engineering*, vol. 5, no. 2, pp. 72-78, June 2019, doi: 10.23919/CJEE.2019.000013.

[10] M. Gaglani, R. Rai and S. Das, "Implementation Of Multilevel Battery Charging Scheme For Lithium-ion Batteries," *2019 National Power Electronics Conference (NPEC)*, 2019, pp. 1-6, doi: 10.1109/NPEC47332.2019.9034748.

[11] D. Xiao *et al.*, "Reduced-Coupling Coestimation of SOC and SOH for Lithium-Ion Batteries Based on Convex Optimization," in *IEEE Transactions on Power Electronics*, vol. 35, no. 11, pp. 12332-12346, Nov. 2020, doi: 10.1109/TPEL.2020.2984248.

[12] T. Takayama and K. Kourai, "Optimization of Parallel Applications Under CPU Overcommitment," *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2020, pp. 51-60, doi: 10.1109/COMPSAC48688.2020.00017.

[13] X. Tang and Z. Fu, "CPU–GPU Utilization Aware Energy-Efficient Scheduling Algorithm on Heterogeneous Computing Systems," in *IEEE Access*, vol. 8, pp. 58948-58958, 2020, doi: 10.1109/ACCESS.2020.2982956.

[14] S. N. Rao, G. Shobha, S. Prabhu and N. Deepamala, "Time Series Forecasting methods suitable for prediction of CPU usage," *2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, 2019, pp. 1-5, doi: 10.1109/CSITSS47250.2019.9031015.

[15] R. Rechter *et al*., "Product Reliability Methods to Enable High Performance CPU's," *2019 IEEE International Reliability Physics Symposium (IRPS)*, 2019, pp. 1-5, doi: 10.1109/IRPS.2019.8720500.

[16] N. Schmitt, J. Bucek, J. Beckett, A. Cragin, K. -D. Lange and S. Kounev, "Performance, Power, and Energy-Efficiency Impact Analysis of Compiler Optimizations on the SPEC CPU 2017 Benchmark Suite," *2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)*, 2020, pp. 292-301, doi: 10.1109/UCC48980.2020.00047.

[17] P. Czarnul and P. Rościszewski, "Auto-tuning methodology for configuration and application parameters of hybrid CPU + GPU parallel systems based on expert knowledge," *2019 International Conference on High Performance Computing & Simulation (HPCS)*, 2019, pp. 551-558, doi: 10.1109/HPCS48598.2019.9188060.

[18] G. Salvadori, F. Leccese, C. Burattini and F. Bisegna, "Photobiological and Circadian Effects of LED Displays: Comparison between Different Laptop Configurations," *2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC /I&CPS Europe)*, 2018, pp. 1-6, doi: 10.1109/EEEIC.2018.8494402.

[19] M. Ye and L. Li, "A Method Suitable for LCD Monitors to Automatically Compensate and Calibrate Brightness," *2019 International Conference on Robots & Intelligent System (ICRIS)*, 2019, pp. 4-7, doi: 10.1109/ICRIS.2019.00011.

[20] Y. Yang and W. Chen, "Parallel Algorithm for Moving Foreground Detection in Dynamic Background," *2012 Fifth International Symposium on Computational Intelligence and Design*, 2012, pp. 442-445, doi: 10.1109/ISCID.2012.270.