ANALYSIS OF POWER OPTIMIZATION TECHNIQUES IN VISION BASED ENVIRONMENT MONITORING SYSTEMS

Umair Ali Khan*, Fareed Hussain Mangi**, Intesab Hussain Sadhayo*

ABSTRACT

Intelligent environment monitoring confronts the major challenge of electrical power minimization with the constraints of real-time response and high quality of service. Such systems are mostly battery-operated and typically perform the tasks of monitoring various (complex) physical phenomena (e.g., forest fire, solar radiation, crowed monitoring, intelligent traffic surveillance, border security, etc) in a hazardous environment and must have adequate processing capabilities to meet the real-time response requirements. With these capabilities being the focus of design, high power consumption is inevitable which is not affordable in the intended environments of operation. A system must be designed with power-efficient components as well as online power optimization strategies. This paper presents a critical analysis of the requirements and constraints for power and resource awareness in environment monitoring systems. We classify the existing power management techniques used in vision-based environment monitoring systems and identify the set of features that serves as a criterion to evaluate the performance of a technique. Our critical analysis and experiments with carefully designed systems demonstrate that a machine-learning based power optimization strategy gives optimal performance in dynamic environments and outperforms other techniques.

Keywords: *Power optimization; environment monitoring; visual sensor networks.*

1. INTRODUCTION

Due to the increased popularity and efficacy of visual sensors over the past few decades, intelligent environment monitoring of hazardous or rough environments is preferably performed using vision-based systems that are mostly battery operated. The systems developed in this context cover a wide range of applications which include monitoring indoor environmental state of large communication places, monitoring human behavior and privacy according to environmental perspective, detecting intrusions, object recognition and tracking, traffic monitoring, etc (to name a few). The advantages of a vision-based system are the following.

(i) Images contain a lot of information that can be processed and manipulated in a number of ways.
(ii) Heterogeneity of visual sensors can be used to perform a variety of tasks by a single system.
(iii) An environmental scene can be analyzed from different perspectives using image processing techniques.
(iv) The use of bulky and expensive visual sensors can be avoided.

However, managing power consumption of the individual visual sensor nodes in a vision-based system, primarily with the goal of extending their lifetime, is widely recognized as a fundamental challenge in intelligent environment monitoring. Since the monitoring systems are usually desired to be highly responsive (i.e., minimum response time at the detection of a particular event), a high processing capability is required, which in turn results in increased power consumption. Therefore, apart from high demands in computing performance, power-awareness is also of particular interest in vision-based intelligent environment monitoring systems.

The power optimization techniques proposed in this context usually fall into two categories: (i) static techniques, such as design-time synthesis and compilation for low-power consumption, and (ii) dynamic techniques that exploit runtime behavior of visual sensors to optimize power consumption [1][2]. These techniques are collectively known as Dynamic Power Management (DPM) that dynamically switches the power states of a system (or its individual components) based on the system's workload history, prediction of the future activity, processing requirements and desired performance.

Power management policies exploit the fact that an environment monitoring system is not always intended to deliver maximum performance, especially when the events occur at low rate or a high quality of service is not required. Therefore, these policies attempt to find the periods in which the system serves no or light workload. Based on this knowledge, some or all components of a system are switched to low-power states.

* Department of Computer Systems Engineering
** Department of Energy & Environment Engineering,
Quaid-e-Awam University of Engineering, Science & Technology, Nawabshah, Pakistan.

A good DPM policy should follow a proactive approach, rather than relying only on a reactive strategy. Based on the current environmental conditions, the DPM policy should anticipate the future usage pattern of a system and should take proactive measures for power management. This extends the scope of DPM from statically optimized resources and power management to an intelligent and adaptive decision-making strategy.

The selection and/or formulation of a DPM policy for a vision-based system is not trivial due to the following challenges.

(i) Multiple power states of different components on a visual sensor node (e.g., processing, idle, sleep, off, etc) have different power consumption and performance levels. Therefore, an optimal tradeoff needs to be found.
(ii) The latency involved in switching among different power states is one of the major issues in a hard real-time environment. Furthermore, the power consumed to switch the power state of a component is also non-negligible. Therefore, it is very important to know when the state-switching latency and the resulting power consumption is effectively compensated and provide overall power savings.
(iii) An accurate prediction of workload results in a more effective state-switching policy, hence reducing the overall latency and increasing power savings [28][29]. Several studies show that the workload of even non-stationary environments exhibit long-term similarity [30][31]. However, accurate workload prediction in dynamic environments to capture the short-term variation is challenging.
(iv) Another challenge is to optimize the data transmission that consumes significant amount of power. When small response time and high quality of service are required, power optimization becomes challenging due to frequent transmissions.

This paper provides an extensive survey of the existing DPM policies in vision-based environment monitoring. We discuss the situations in which a specific DPM policy performs well and the factors that lead to the degradation of a DPM policy. Since real environments are highly dynamic, an important aspect of our investigation is to highlight the DPM policies that can deal with non-stationary request patterns. We identify several performance parameters that define the optimal behavior of a DPM policy. Using these parameters as evaluation criteria, we justify why static power management policies or those requiring a specific model of the system may not always lead to optimal behavior. This rigorous analytical survey and the deduced conclusions lead us to highlight the potential strategy for the DPM of a vision-based system running in a highly dynamic environment. We justify why the optimal behavior in a dynamic environment can be achieved only with an online machine-learning based DPM policy.

Figure 1 shows our vision-based traffic monitoring system based on heterogeneous visual sensors. The system performs online power optimization using machine-learning algorithms.

Figure 1: Our vision based traffic monitoring system under power management [32]

2. DPM REQUIREMENTS

Independent of the type of environment a system is intended to operate in, formulating and implementing a DPM problem for the system is not trivial. A DPM problem requires taking into consideration a number of factors that play critical roles in the success of a DPM policy. In general, a good DPM policy must consider the following factors.

2.1 Identifying the Culprit Components

In most cases, it is not necessary to target the whole system for power management. Instead, the power consumption can be optimized by targeting specific component(s) of the system. For example, in most of the mobile computing devices (e.g., cell phones), the display is the most power-consuming component [33]. A proper power management of the display component results in a significant power savings in the overall system. Therefore, before implementing a DPM policy, the most power-hungry elements in a system must be identified.

2.2 Environmental Uncertainty

The request processing pattern of a complex system is usually highly dynamic and dependent on the application, input data and the user context. The variations in the workload bring about changes in the system's usage pattern and substantially influence the power consumption and performance. Therefore, an important feature of a DPM policy is to learn the environment, adapt to different hardware systems, and operate under different working environments [34].

2.3 Power-Performance Tradeoff

Minimizing power consumption results in an inevitable performance degradation in every computing system.

Power optimization becomes a challenge for the systems where a real-time response is required. However, for the systems that are required to deliver a soft real-time response, performance can be traded for low power consumption. For such systems, a number of optimal solutions corresponding to different power-performance settings exist. A good DPM policy should allow the user to flexibly control the power-performance tradeoff to select one of these optimal solutions.

2.4 Transition Costs

Switching a device into low power or non-operational mode and then switching it back to an active (operational) mode results in a sizable and non-negligible amount of power consumption as well as latency. A DPM policy that tends to switch off a device too frequently during the periods of high activity, in fact results in extra power consumption and performance degradation. Therefore, a DPM policy should be able to find an appropriate balance among the state transitions. The decisions of switching the power states of a system should be taken only when the resulting power-consumption and latency can amortize the power savings achieved by the state transition.

2.5 Processing Requirements, Deadlines and Preferences

A DPM policy should be able to identify the processing requirements of individual tasks submitted to the system. Some requests may have stringent deadlines and should be processed with highest priority. In such cases, power consumption can be sacrificed to deliver a real-time response. Likewise, during different periods of the system's operation, the processing requirements and performance preferences may change. During the peak hours of a system's operation, the DPM policy should prefer high performance at the expense of higher power consumption.

3. DYNAMIC POWER MANAGEMENT TECHNIQUES

Several power management techniques have been studied for environment monitoring systems that can be broadly classified into two categories: adaptive (online) and non-adaptive (static). Essentially, all approaches determine how long a system should remain in an idle state before transiting to a low power (or off) state. Non-adaptive approaches statically determine the idle periods based on the properties of the system and do not consider the arrival rate of requests. Whereas, adaptive power management approaches compute the idle periods dynamically based on the history of incoming requests [35]. Adaptive power management approaches tend to perform better by making appropriate decisions over time based on the change in the load of the system [36]. Figure 2 provides a classification of existing power management techniques for environment monitoring systems.

Before analyzing the existing DPM policies individually, we prefer to present a generic DPM model shown in Figure 3. A particular objective to present this generic DPM model is to aggregate all the necessary components in a single (conceptual) model and then mapping the existing DPM approaches to this model. This helps us to highlight the salient features a particular DPM policy lacks or possesses. The conceptual model presented in Figure 3 depicts a system where the requests can be buffered before processing and an appropriate power-performance criterion can be selected specific to the environment and/or the system. The service requestor in Figure 3 is an image visual sensor that captures the events and buffers them in the service queue. The processing device, referred to as the service provider in this model, retrieves the requests from the service queue and processes them. The power manager is the DPM policy which switches the power modes of the service provider based on the observations received from all the components.

3.1 Greedy Policy

The simplest power management policy is to switch a device to a low-power state as soon as it is idle [37]. The greedy policy does not analyze the past workload of the system and the additional power and/or latency resulted by frequent state-transitions in case of higher workloads.

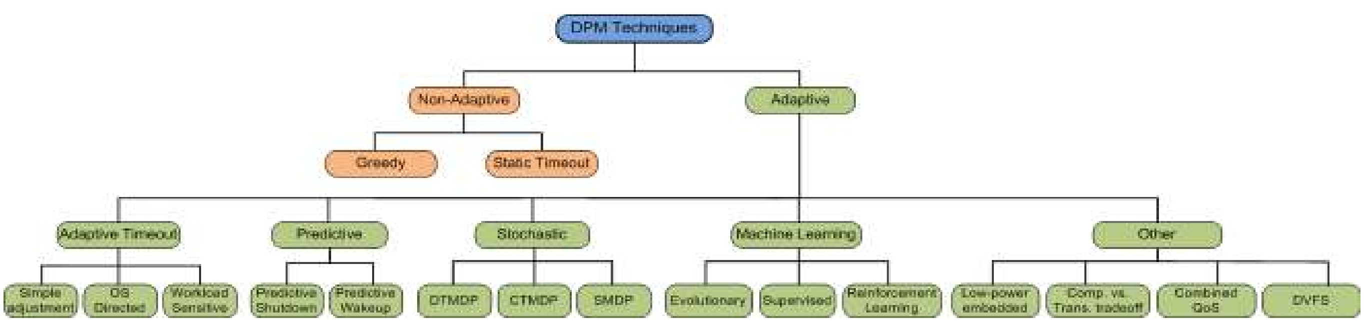

Figure 2: Classification of dynamic power management

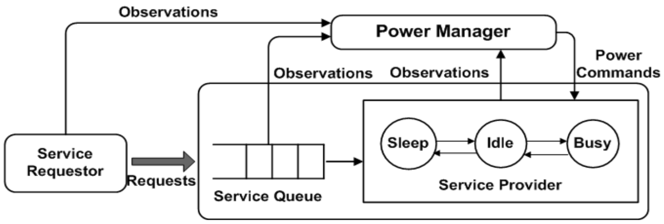

Figure 3: A generic model for power management

As long as the requests' inter-arrival times are long, the greedy policy performs well, as it can amortize the infrequent state-transitions cost in this case. However, real input patterns are usually non-stationary and it is not efficient to turn off a device during a bursty usage pattern. A DPM policy must have a mechanism to detect a period of inactivity to change the power mode of the device.

3.2 Timeout Policies

Instead of an immediate shutdown (or switching a device to low-power mode as soon as it is idle), the timeout policies assume that if a device is idle for a certain period of time, it would remain idle for the same period of time in future and it is relevant to switch it to low-power mode. The advantage of the timeout policy over the greedy policy is that the additional cost of power consumption and latency during the state transitions can be avoided when the workload is high. The two variants of timeout policies are static and adaptive policies [38]. A user defined fixed timeout is used in static policies. Due to their simplicity, static timeout policies are widely used in many commercial appliances, e.g., mobile phones, monitors and hard disks. An adaptive timeout scheme is more effective because it adjusts the timeout threshold according to the system's workload history.

Without incorporating an accurate workload prediction, timeout policies turn out to be too naïve. Once the systems becomes idle and starts a timer, it consumes power until the timeout expires. Furthermore, these policies are implementable only at the idle (active) state of a system and cannot deal with non-operational or sleep states. Additionally, timeout policies lack the mechanism for controlling the power-performance tradeoff.

Figure 4 shows the mapping of greedy and timeout policies to the generic DPM model shown in Figure 3.

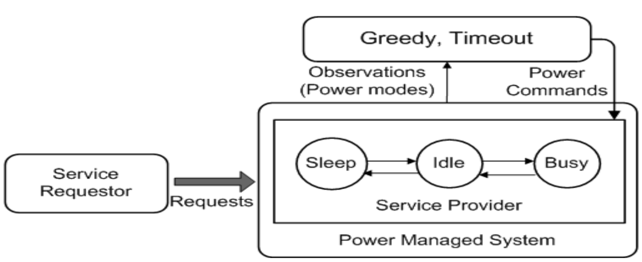

Figure 4: Mapping of greedy and timeout policies to the generic DPM model.

3.3 Predictive Policies

Predictive policies analyze the history of an environment monitoring system's workload and attempt to predict the future idle periods of the system using statistical approaches. The predictive techniques are extensively researched and used for a number of applications [2][5][6][10][11][12]. The basic approach of all the predictive policies is to use the correlation between the past usage pattern of a system and predict the arrival time of the next event. These policies predict the length of an idle period before it starts. If an idle period is predicted to be longer than a certain threshold (called break-even time), the device enters into a low power state right after it is idle. In order to predict the idle periods, several techniques have been introduced. One such technique uses adaptive learning trees [10] to encode the sequence of idle periods into tree nodes. This policy predicts the length of an idle period with finite-state machines. If an idle period is predicted to be longer than a given break-even period, the confidence level increases; otherwise, the confidence level decreases [2].

Some other techniques use mathematical predictors on the history of system's workloads in order to predict the future workload. The simplest of such predictor uses the weighted average workload filter that predicts the future idle period based on the average of past idle periods. A better variant of this technique is the moving average filter.

Exponential weighted average of past idle periods to predict the future period of inactivity is also studied. A better approach uses least mean square filter that works as an adaptive filter whose coefficients are modified based on the prediction error. In this sense, it works better than the exponential weighted averaging filter [5].

For the environment monitoring systems where the visual sensor nodes have a high degree of uncertainty in the observation, computation and communication, predictive policies do not perform impressively. Moreover, these policies cannot control the power-performance tradeoff. Figure 5 shows the mapping of predictive policies to the generic DPM model.

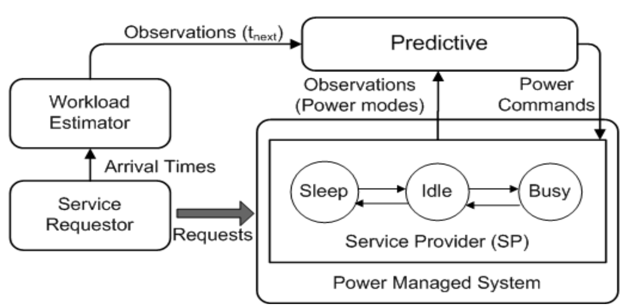

Figure 5: Mapping of predictive policies to the generic DPM model.

3.4 Stochastic Policies

Stochastic policies are based on the probabilistic model of a system's behavior with the associated uncertainty related to the system's inputs. This mathematical model is then used to formulate an optimization problem that is solved during the optimization process to obtain an optimal DPM strategy [3]. The modeling of power states of a device, transition mechanism, and its service queue is generally performed using Markov Decision Process (MDP). An MDP represents a system in which the next state is independent of the history of state transitions and depends only on the current state [9]. Considering the uncertainty in the environment, communication, computation load, occurrence of events, and nature of events in environment monitoring systems, several variants of the stochastic approaches can be used to describe the system and find the optimal DPM policy. These variants include discrete-time, continuous-time and semi-Markov decision processes (represented by DTMDP, CTMDP and SMDP, respectively in Figure 2).

A simple 2-state MDP is shown in Figure 6 where the system has two power states, i.e., active and sleep. The arrows represent the state transitions. Every transition is assigned a certain probability with which the power manager decides to change the state of the system or to remain in the same state.

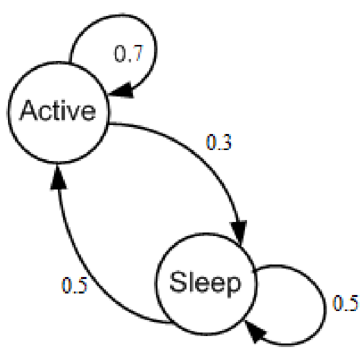

Figure 6: A 2-state Markov decision process

Stochastic techniques support the request queuing as described in the generic model presented in Figure 3 and hence are able to trade power with performance. However, stochastic techniques require the mathematical formulation of the system's model (as shown in Figure 6) which is not only nontrivial, but also makes it environment specific.

3.5 Low-Power Embedded Designs

Low-power embedded designs are also of particular interest in environment monitoring systems. These systems comprise low- or medium resolution image visual sensors (mostly CMOS) integrated with an embedded computing platform [13][14][15][16][17]. The key power management strategy in such systems is based on system-level dynamic power management because embedded systems provide good power management in terms of better control and access to various system components like system buses, memory, communication units and power modes of processor and other components. The power management policy in such systems is usually implemented at operating system level that deals with changing the power modes of individual system components based on the occurrence of events, available resources and/or processing requirements. Since the vision-based environment monitoring usually requires high-performance on-board image and video processing, these systems also use dedicated FPGAs or DSPs to perform image acquisition, compression, encoding and decoding operations. A generic representation of such systems is shown in Figure 7.

Unlike the additional DPM strategies implemented at application level, the DPM techniques applied at the OS level usually do not deliver aggressive power savings. Although such systems usually prove to be highly responsive, power savings are not significant.

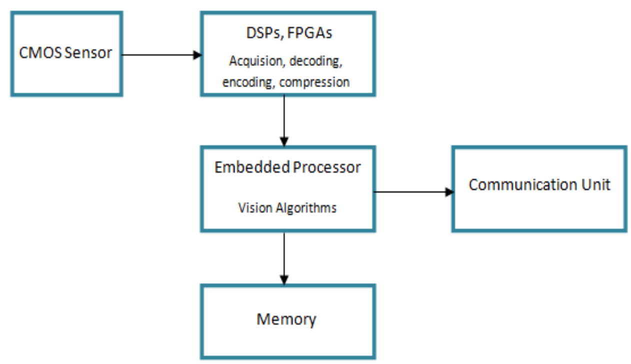

Figure 7: Generic representation of surveillance systems based on embedded design

3.6 Computational load and transmission power trade-off

Data communication in vision-based environment monitoring systems is more power consuming than processing and sensing. Therefore, significant power savings can be obtained by minimizing the frequency of transmission [18][19]. Some approaches focus on finding an appropriate tradeoff between processing and communication. In order to reduce the transmission load, one of the solutions is to perform the local data/image compression. However, the power required for image compression is comparable to the one consumed in transmission. In [18], the compression rates, processing times and power consumption of different compression algorithms are studied. However, the performance of these algorithms heavily depends on the application and the overall visual sensor configuration. This approach can be implemented just as a part of the overall dynamic power management strategy and requires a detailed investigation of the power consumptions and processing load of different compression algorithms for a particular setup.

3.7 Quality of Service (QoS) Based DPM

This technique combines power optimization with QoS adaptation [20][21]. The degree of freedom in adapting QoS parameters strictly depends on its designated application. If it is acceptable for the user and non-critical in the context of the application, the level of QoS may be manipulated and power can be saved. Thus, this method is a tradeoff between the QoS and power consumption.

This technique dynamically changes the power modes of individual components (e.g., visual sensors, processing units, network devices) based on the required QoS level. The system runs in normal mode when no event is detected and hence delivers minimum QoS level with maximum power savings. At the detection of an event, adequate QoS parameters are selected for each component. However, this approach does not take into account the latency involved in switching the power modes of different components. Since this approach is aimed to implement on a hard real-time system, this latency cannot be ignored.

3.8 Dynamic Voltage and Frequency Scaling (DVFS)

Another wide area of research in dynamic power management is dynamic voltage and frequency scaling. DVFS exploits the fact that the peak clock frequency of a processor is proportional to the supply voltage, while the amount of dynamic power required for a given workload is proportional to the square of the supply voltage [22]. DVFS varies the clock frequency of the processor and its supply voltage in such a way that the requests are processed within their deadlines or with minimum latency. This requires a proper scheduling of processor's clock frequency and supplied voltage. The task of scheduling processor's frequency and voltage is performed by the operating system that has a global view of the resource usage, requirements and workload. A number of techniques have been proposed for predicting the processor's operating frequency and voltage at the operating system level [23][24][25][26][27]. The rationale in all these techniques is to analyze the processor's workload history and to predict the future workload in order to set an appropriate frequency and voltage. However, DVFS cannot control the power-performance tradeoff.

3.9 Machine-learning Based Policies

Machine-learning based techniques are of particular interest to find patterns of a system's usage and its workload. Predicting a system's periods of inactivity is also performed using genetic (evolutionary) algorithms [39]. This approach exploits the relationship between adjacent periods of activity and inactivity to predict the future period of inactivity and decides if it is appropriate to switch the system to low-power state. Nevertheless, this approach works no better than the previously described predictive policies.

Some supervised learning techniques for power management [40][41] use offline data analysis to extract essential features of the system which collectively describe the power profile of the system. The system is trained with these features to generate a classifier (e.g., Bayesian) to predict the system's performance state from new observations and then using this predicted state to look up the optimal power management action from a pre-computed policy table. The offline data collection, analysis, and identification of the essential features is cumbersome and requires the same effort as in the case of stochastic policies.

A variant of unsupervised learning, called Reinforcement Learning (RL), is found to be significantly more effective to learn a system's environment and take better actions for power optimization. Power optimization with RL does not require a priori model of the system. Instead, the main theme is to learn the environment of a system by starting with no or little knowledge and trying out various alternatives in each state of the system until good actions for each state are learnt.

RL is originally applied to MDPs and an power optimization problem does not essentially represent an MDP. However, some model-free, RL based DPM approaches proposed in [42][43] prove that RL can be successfully applied to DPM problems. RL is first implemented for DPM in [42], where some pre-selected policies, referred to as experts, are selected under different operating conditions for power optimization. Each expert is assigned a weight that varies throughout the online learning process and represents the effectiveness of an expert in a given set of conditions. In each state of the system, an expert with the highest weight is selected with the highest probability. This approach does provide an optimal power management technique, but its performance is strongly dependent to the selected experts.

Another model-free, RL-based DPM technique is proposed in [43] that does not require a set of experts and is able to effectively control the power-performance tradeoff. Nevertheless, this technique works with a huge state-space and is based on a discrete-time stochastic process. Hence, it is not memory- and computing-efficient.

The authors in [44] propose an improved, continuous-time RL based policy by incorporating a workload prediction.

A naïve Bayes classifier predicts the inter-arrival times of future events by the history of past inter-arrival times. However, the implementation of this learning policy is done on a wireless LAN card that sends/receives data from the protocols generating regular traffic having higher degree of predictability. This workload prediction cannot deliver acceptable accuracy in non-stationary environments.

We address this issue in [45][46] using the DPM model shown in Figure 3. We specifically target the non-stationary workloads with a machine-learning approach that does not require any prior model. We use a multi-layer neural network with back-propagation algorithm to provide estimated workload information to the learning algorithm. With the predicted workload information, this approach selects appropriate timeout policies in the idle state of a system. The wakeup decision in sleep state is made on the basis of service queue populated with certain number of requests. In contrast to the existing DPM approaches that focus on an immediate response from the service provider as soon as there is a request in service queue, we opt to postpone the request processing to explore a deeper power-performance tradeoff. The time spent by the system in active and inactive states is controlled by an adjustable power-performance tradeoff parameter. Workload prediction using a multi-layer neural network achieves higher accuracy with the non-stationary data and the algorithm is capable of exploring the tradeoff in the power-performance design space. However, this approach does not ascertain the upper and lower bounds of average latency in request processing and also results in high latency when the workload is low.

We further address this issue in [28][29] and propose an RL based policy called Online Learning of Timeout Policies (OLTP) for the power management of environment monitoring systems. This policy targets both the idle and sleeps states to give a better power-performance tradeoff. Additionally, we extend the scope of OLTP by enabling it to handle multiple sleep and idle states of the modern computing systems. Our experimental results show that a machine-learning based policy is able to learn the optimal timeout values for each state of the system based on the environmental changes and system's workload. Our rigorous comparative analysis also shows that the OLTP outperforms existing DPM policies for environment monitoring.

Figure 8 shows the mapping of stochastic and machine-learning policies to the generic DPM model. Through our rigorous analysis, we identify a set of distinct features that serves as a criterion to evaluate the performance of a given DPM policy. The comparative analysis of the existing DPM policies for environment monitoring with respect to these features is given in Table 1. A YES or NO value followed by a (+) sign represents the merit of a DPM policy for the corresponding feature. It is evident that the machine-learning based DPM policy outperforms other policies.

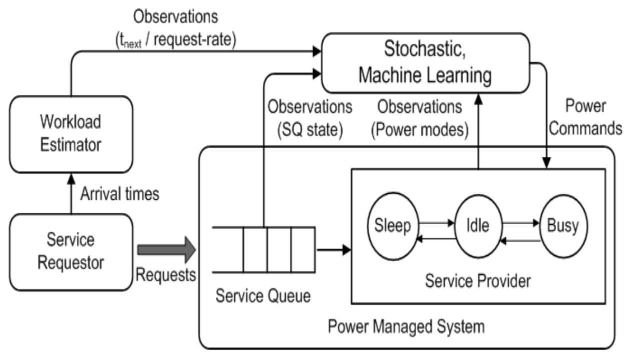

Figure 8: Mapping of stochastic and machine-learning policies to the generic DPM model.

4. CONCLUSION

In this paper, we performed a critical analysis of the power management techniques for vision-based environment monitoring systems. We described some essential environmental and system-level factors an power management policy should be focused on. Subsequently, we provided a detailed analysis of the existing power management policies and argued why a RL-based DPM approach can outperform other policies. An important part of our discussion involved the investigation of the applicability of the existing DPM policies to the generic DPM model described in Section 3. Furthermore, we highlighted essential features of a good DPM policy and compared the existing DPM policies with respect to the identified performance parameters.

REFERENCES

[1] Arnold Maier, "Dynamic power aware camera configuration in distributed embedded surveillance clusters", PhD Dissertation, Technical University Graz.

[2] Y. Lu, G. Micheli, "Comparing System-level power management policies", IEEE conference on Design & Test of Computers, 2001

[3] A. Fallahi, E. Hossain, "QoS provisioning in wireless visual sensor networks", IEEE journal of wireless communication, vol.14, No.1, pp.40-49, 2007.

[4] B. Brock, K. Rajamani, "Dynamic power management for embedded systems", IEEE SOC Conference, pp. 1-25, 2003.

[5] A. Sinha, A. Chandrakasan, "Dynamic power management in wireless visual sensor networks", IEEE Design & Test of Computers, vol. 18, No. pp. 62-74, 2001.

[6] L. Benini, A. Bogliolo, G. D. Micheli, "A survey of design techniques for system-level dynamic power management", IEEE Transactions on very large scale integration (VLSI) systems, vol.8, No.3, 2000.

[7] G. Norman, D. Parker, M. Kwiatkowska, "Using probabilistic model checking for dynamic power management", Formal aspects of computing, vol. 17, No. 2, pp. 160-176, Springer-Verlag, 2005.

[8] Q. Qiu, Q. Wu, M. Pedram, "Stochastic modeling of a power-managed system – construction & optimization", IEEE Transactions on computer-aided design of integrated circuits & systems, vol.20, No.10, pp. 1200-1217, 2001.

[9] Guo, Xianping, and Onésimo Hernández-Lerma, "Continuous-time Markov decision processes", Springer Berlin Heidelberg, 2009.

[10] E. Y. Chung, L. Benini, G. D. Micheli, "Dynamic power management using adaptive learning tree", Proceedings of the 1999 international conference on computer aided design, pp. 274-279, 1999.

[11] T. Simunic, L. Benini, P. Glynn, G. Micheli, "Dynamic power management for portable systems", Proceedings of the 6th annual international conference on mobile computing and networking, MobiCom, pp. 11-19, 2000.

[12] C. Hwang, A.C. Wu, "A predictive system shutdown method for power saving of event-driven computation", ACM Journal on design automation of electronic systems, vol.5, No. 2, pp. 226-241, 2000.

[13] Z. Y. Cao, Y. Ji, M. Z. Hu, "An image Visual sensor node for wireless visual sensor networks", Proceedings of the International Conference on Information Technology: Coding and Computing", pp. 740-745, 2005.

[14] M. Bramberger, R. P. Pflugfelder, B. Rinner, "A smart camera for traffic surveillance", Proceedings of the first workshop on Intelligent Solutions in Embedded Systems", pp. 153-164, 2003.

[15] C. B. Margi, R. Manduchi, K. Obraczka, "Power consumption tradeoffs in visual sensor networks", 12th Brazillian Symposium on Computer Networks, pp. 1-4, 2006.

[16] B. Brock, K. Rajamani, "Dynamic power management for embedded systems", IEEE International SoC conference, pp. 1-25, 2003.

[17] W. Feng, E. Kaiser, M. Baillif, "Panoptes: scalable low-power video visual sensor networking technologies", ACM Transactions on multimedia computing, communications and applications, vol.1, No.2, pp. 151-167, 2005.

[18] L. Ferrigno, S. Marano, A. Pietrosanto, "Balancing computational and transmission power consumption in wireless image visual sensor networks", IEEE Conference on virtual environments, human-computer interfaces, and measurement systems, pp. 1-6, 2005.

[19] V. Raghunathan, S. Park, M. B. Srivastava, "Power-aware wireless micro-visual sensor networks", IEEE Signal Processing Magazine, vol. 19, No. 2, pp. 40-50, 2002.

[20] A. Maier, B. Rinner, T. Trathnigg, "Combined dynamic power and QoS management in embedded video surveillance systems", Proceedings of the second workshop on intelligent solutions in embedded systems, pp. 63-77, 2004.

[21] A. Maier, "Dynamic power aware camera configuration in distributed embedded surveillance clusters", PhD Dissertation, Technical University Graz, 2006.

[22] Mudge. T, Ann Arbor, "Power: a first-class architectural design constraint", Computer, Vol. 34, No. 4, pp. 52-58, 2001.

Table 1: Comparative analysis of power management techniques in environment monitoring systems

	Performance constraint	Performance tradeoff	Workload handling	Queuing	Model-free	Non-stationarity	Offline analysis	Complexity
Static timeout	No (-)	No (-)	No (-)	No (-)	Yes (+)	No (-)	No (+)	Low
Adaptive timeout	No (-)	No (-)	Yes (+)	No (-)	Yes (+)	No (-)	No (+)	Low
Predictive shutdown	No (-)	No (-)	Yes (+)	No (-)	Yes (+)	Yes (+)	No (+)	Low
Predictive wakeup	Yes (+)	No (-)	Yes (+)	No (-)	Yes (+)	Yes (+)	No (+)	Low
Stochastic	Yes (+)	Yes (+)	Yes (+)	Yes (+)	No (-)	Yes (+)	Yes (-)	High
Low-power embedded design	N/A	No (-)	No (-)	No (-)	N/A	N/A	N/A	low
Computing-Transmission tradeoff	No (-)	No (-)	No (-)	N/A	N/A	N/A	N/A	high
QoS tradeoff	Yes (+)	Yes (+)	No (-)	No (-)	N/A	No (-)	N/A	high
DVFS	Yes (+)	No (-)	Yes (+)	No (-)	Yes (+)	No (-)	No (+)	low
Evolutionary	No (-)	No (-)	Yes (+)	No (-)	Yes (+)	No (-)	No (+)	Low
Supervised learning	Yes (+)	Yes (+)	Yes (+)	No (-)	Yes (+)	Yes (+)	Yes (-)	Moderate
Online RL	Yes (+)	Yes (+)	Yes (+)	Yes (+)	Yes (+)	Yes (+)	No (+)	Moderate

[23] D. Grunwald, P. Levis, K. Farkas, "Policies for Dynamic Clock Scheduling", Proceedings of the 4th conference on symposium on operating system design & implementation, pp. 1-6, 2000.

[24] K. Flautner, T. Mudge, "Vertigo: Automatic performance-settings for Linux", Proceedings of the 5th symposium on operating systems design and implementation, pp. 47-52, 2002.

[25] T. Pering, T. Burd, R. Brodersen, "The simulation and evaluation of dynamic voltage scaling algorithms", Proceedings of the international symposium on low power electronics and design, pp. 76-81, 1998.

[26] K. Govil, E. Chan, H. Wasserman, "Comparing algorithms for dynamic speed-setting of a low-power CPU", Proceedings of the 1st annual international conference on mobile computing and networking, pp. 13-25, 1995.

[27] M. Magno, A. Lanza, "Power aware multimodal embedded video surveillance", IEEE conference on VLSI system on chip, pp. 264-279, 2010.

[28] Khan, Umair Ali, and Bernhard Rinner, "Online learning of timeout policies for dynamic power management." ACM Transactions on Embedded Computing Systems, vol. 13, No. 4, pp. 1-25, 2014.

[29] U. A. Khan, F. A. Jokhio, I. H. Sadhayo, "Reinforcement Learning for dynamic power management of embedded visual sensor nodes", Mehran University Research Journal of Engineering & Technology, vol. 33, No.2, 2014.

[30] P. Bogdan and R. Marculescu, "Cyber physical systems: Workload modeling and design optimization", IEEE Transactions on Design & Test, vol. 28, No. 4, pp. 78–87, 2011.

[31] P. Bogdan and R. Marculescu, "Non-stationary traffic analysis and its implications on multi-core platform design", IEEE Transactions on Computer Aided Design of Integrated Circuits Systems, vol. 30, No. 4, pp. 508–519, 2011.

[32] U. A. Khan, B. Rinner, "A reinforcement learning framework for dynamic power management of a portable, multi-camera traffic monitoring system", In Proceedings of IEEE conference on Green Computing & Communication (GreenCom), pp. 557-564, Besancon, France, 2012.

[33] J. Lorch and A. Smith, "Software strategies for portable computer power management", IEEE Transactions on Personal Communications, vol. 5, No. 3, pp. 60–73, 1998.

[34] W. Liu, Y. Tan, and Q. Qiu, "Enhanced Q-learning algorithm for dynamic power management with performance constraint", In Proceedings of the Conference on Design, Automation and Test in Europe, pp. 602–605, 2010.

[35] D. Ramanathan, S. Irani, and R. Gupta, " An analysis of system level power management algorithms and their effects on latency", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 21, No. 3, pp. 291–305, 2002.

[36] M. Pedram, "Power aware design methodologies", Springer, 2002.

[37] M. Pedram, "Power optimization and management in embedded systems", In Proceedings of the ASP-DAC Design Automation Conference, pp. 239–244, 2001.

[38] Y. Lu and G. De Micheli, "Comparing system-level power management policies", IEEE Transactions on Design and Test of Computers, vol. 18, No. 2, pp. 10–19, 2001.

[39] K. Fei, T. Pin, Q. Shi, and L. Xiao, "Genetic algorithm based idle length prediction scheme for dynamic power management", In Proceedings of IMACS Multi-conference on Computational Engineering in Systems Applications, pages 1437 –1443, 2006.

[40] S. Mannor et al., "Machine-learning for adaptive power management", Autonomic Computing, vol. 10, No. 4, pp. 299–312, 2006.

[41] J. Hwisung and M. Pedram, "Supervised learning based power management for multi-core processors", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 29, No. 9, pp. 1395 –1408, 2010.

[42] G. Dhiman and T. Rosing, "Dynamic power management using machine-learning", In Proceedings of IEEE/ACM International Conference on Computer-Aided Design, pp. 747–754, 2006.

[43] Y. Tan, W. Liu, and Q. Qiu, "Adaptive power management using reinforcement learning", In Proceedings of International Conference on Computer-Aided Design, pp. 461–467, 2009.

[44] Y. Wang, Q. Xie, A. Ammari, and M. Pedram, "Deriving a near-optimal power management policy using model-free reinforcement learning and bayesian classification", In Proceedings of Design Automation Conference, pp. 41–46, 2011.

[45] U. Khan and B. Rinner, "Dynamic power management for portable, multi-camera traffic monitoring", In Proceedings of the 18th IEEE Real-Time and Embedded Technology and Applications Symposium (WiP), pp. 37–40, 2012.

[46] U. Khan, M. Godec, M. Quaritsch, M. Hennecke, H. Bischof, and B. Rinner, "Mobitrick: Mobile traffic checker", In proceedings of ITS World Congress, pp. 1-10, 2012.