2008 International Symposium on Nonlinear Theory and its Applications
NOLTA'08, Budapest, Hungary, September 7-10, 2008

Fair Resource Allocation for Energy and QoS Trade-off Management in Battery-Driven Real-Time Systems

Fumiko Harada†, Toshimitsu Ushio‡, Yukikazu Nakamoto*

†College of Information Science and Engineering, Ritsumeikan University
Kusatsu, Shiga, 525-8577, Japan
‡Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, 560-0043, Japan
* Graduate School of Applied Informatics, University of Hyogo
Harborland Center Bldg. 22–23F, 1-3-3 Higashikawasaki, Chuo-ku, Kobe, Hyogo, 650-0044, Japan
Email: harada@cs.ritsumei.ac.jp, ushio@sys.es.osaka-u.ac.jp, nakamoto@ai.u-hyogo.ac.jp

Abstract—In battery-driven real-time systems, there is requirement for adapting both energy consumption and Quality of Service (QoS) of tasks according to their importance. This paper proposes a power-aware resource allocation method to resolve a trade-off between the energy consumption and QoS according to their importance with guaranteeing the fairness. The proposed method accounts for energy consumption by devices as well as CPU with DVS and guarantees the system lifetime requirement. We formulate the resource allocation problem as a nonlinear optimization problem and propose a method to find the optimal allocation by an iterative method.

1. Introduction

Several wireless sensor networks (WSN) have the lifetime requirements that the lifetime of each battery-driven sensor node is guaranteed to be longer than a given one. By saving the energy consumption in each node, the lifetime can extend as long as possible. On the other hand, saving the energy can force smaller sampling rate to keep the schedulability. Reduction of the sampling rate negatively affects on the Quality of Service (QoS) of the result of the data processing. The importance of saving the energy consumption and guaranteeing the QoS changes occasionally according to the residual energy, the longness of the lifetime, and so on. Thus, the resource allocation in a node should consider their importance.

Several researches have been studied to resolve the trade-off [1, 2]. In [1], a cross-layer energy and QoS adaptation framework has been proposed, which coordinates the resource allocation based on a given trade-off scheme. However, it does not consider the occasional change of trade-off scheme by the residual energy, the lifetime requirement, and so on.

In this paper, we propose a resource allocation method to resolve the trade-off between the energy and the QoS under the lifetime requirement with considering the occasionally variant importance. The proposed method adapts the tasks' frequencies and the CPU speed under the lifetime and fairness constraints. Though it is suitable for sensor nodes, it can be applied to general soft real-time systems with DVS-enabled CPUs and other devices such as wireless communication devices.

The rest of this paper is organized as follows. Section 2 gives the real-time system model and describes the resource allocation architecture. Section 3 formulates the resource allocation problem and discuss the optimality of it. A simulation result is shown in Section 4. Finally, Section 5 concludes this paper.

2. Resource allocation for the trade-off management

2.1. System model

Consider an ideal CPU with the DVS capability. Its available CPU speeds are denoted by $\{s_1, s_2, \ldots, s_m\}$, where $s_1 < s_2 < \cdots < s_m$. The power consumption for s is given by Ks^3, where $K > 0$ is known.

The task set $\{\tau_1, \ldots, \tau_n\}$ is independent and periodic. Each task τ_i is represented as a 6-tuple $(x_i^{\min}, x_i^{\max}, c_i^{\mathrm{CPU}}, c_i^{\mathrm{dev}}, e_i^{\mathrm{dev}}, \phi_i)$. $x_i^{\min}$ and $x_i^{\max}$ are the minimum and maximum task frequency, respectively. The word "frequency" means (1/(period)) in this paper. The execution of a job of τ_i is divided into the execution on the CPU and that on the other devices. c_i^{CPU} is the CPU cycle demand for the execution on the CPU. c_i^{dev} is the execution time on the devices. For a CPU speed s_j, the execution time of a job of τ_i is given by $\frac{c_i^{\mathrm{CPU}}}{s_j} + c_i^{\mathrm{dev}}$. e_i^{dev} is the energy consumption for execution on the devices. The QoS of τ_i varies according to its frequency. The resource consumption function $\phi_i : [0, 1] \to [x^{\min}, x^{\max}]$ represents the required frequency to execute a job of τ_i with a value of QoS.

2.2. Resource allocation architecture

In several battery-driven real-time systems, the importance of saving the energy consumption and guaranteeing the QoS may be occasionally variant. Their importance may be determined according to the residual energy, the

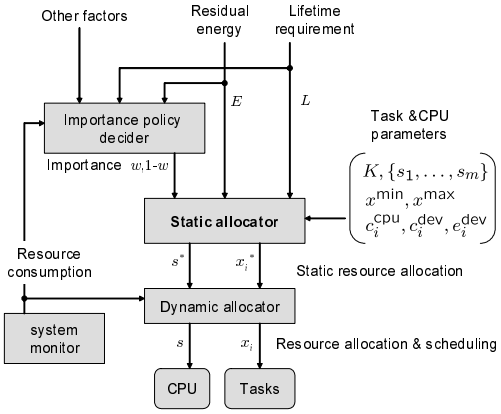


Figure 1: The resource allocation architecture.

lifetime requirement, and other factors such as the external environment. For example, when the residual energy is enough large, a resource allocation with larger energy consumption is acceptable to guarantee high QoS. On the other hand, an allocation with reasonably small energy consumption and a reasonably high QoS is desirable if the residual energy of the device is moderate. If the required lifetime is enough long, saving the energy is more important than guaranteeing the QoS. For another example in WSNs, the sensor data implying an emergency enhances importance of guaranteeing the QoS. Thus, we introduce a resource allocation to resolve the trade-off between saving the energy consumption and guaranteeing the QoS according to their current importance.

Shown in Fig. 1 is the resource allocation architecture for the energy and QoS trade-off management, which consists of the *importance policy decider*, *static allocator*, *dynamic allocator*, and *system monitor*. The input parameters to the importance policy decider are the residual energy E, lifetime requirement L, current resource consumption, and other factors which affect on the importance. The decider determines the relative importance parameter w of saving the energy compared with guaranteeing the QoS according to these parameters. For example, it produces a small value of w for the sufficiently large residual energy while it does a large value of w for the small residual energy. w is sent to the static allocator. According to w, the static allocator calculates each task's optimal frequency x_i^* and the optimal CPU speed s^* to resolve the trade-off between the energy consumption and QoS under the lifetime requirement and the energy limitation. The calculation is performed based on w, L, E, and the input tasks' and CPU's parameters. Each task releases its jobs according to the adapted frequency x_i^*. The dynamic allocator sets the CPU speed to s^* and schedules the released jobs with a scheduling algorithm such as Earliest Deadline First or Rate Monotonic[3]. In execution of a released job, the CPU cycle demand and achieved QoS dynamically change. The dynamic allocator performs the resource reclaiming and adapts the periods and the CPU speed dynamically. This architecture enables to resolve the trade-off between saving the energy consumption and guaranteeing the QoS based on their occasionally variant importance.

In this paper, we focus on the static allocator in detail. Section 3 describes what resource allocation problem is considered and how the optimal allocation is determined in the static allocator.

3. Static allocator

3.1. Problem formulation

The static resource allocator solves a problem to find the optimal static allocation to resolve the trade-off. We consider the resource allocation problem as Problem 1.

Problem 1 *$w \in [0,1]$, E, L, U, K, $\{s_1,\ldots,s_m\}$, T, $x_i^{\min}$, $x_i^{\max}$ and ϕ_i are given. Find the optimal values of the frequencies x_i $(i = 1,2,\ldots,n)$ and the CPU speed s for Eqs. (1)–(8):*

$$\max \quad w\frac{\overline{E} - T\sum_{i=1}^{n} x_i(Kc_i^{\mathrm{CPU}}s^2 + e_i^{\mathrm{dev}})}{\overline{E} - \underline{E}} + (1-w)Q \quad (1)$$

$$\text{s.t.} \quad \overline{E} = T\sum_{i=1}^{n} x_i^{\max}(Kc_i^{\mathrm{CPU}}s_m^2 + e_i^{\mathrm{dev}}) \quad (2)$$

$$\underline{E} = T\sum_{i=1}^{n} x_i^{\min}(Kc_i^{\mathrm{CPU}}s_1^2 + e_i^{\mathrm{dev}}) \quad (3)$$

$$x_i = \phi_i(Q),\ i = 1,\ldots,n \quad (4)$$

$$\sum_{i=1}^{n} x_i\left(\frac{c_i^{\mathrm{CPU}}}{s} + c_i^{\mathrm{dev}}\right) \le U \quad (5)$$

$$\sum_{i=1}^{n} Lx_i\left(Kc_i^{\mathrm{CPU}}s^2 + e_i^{\mathrm{dev}}\right) \le E \quad (6)$$

$$x_i^{\min} \le x_i \le x_i^{\max},\ i = 1,\ldots,n \quad (7)$$

$$s \in \{s_1, s_2, \ldots, s_m\} \quad (8)$$

Equation (1) represents the trade-off between the energy consumption and the fair QoS based on the weight parameter w. Its second term is the achieved QoS Q while the first is the normalized total energy consumption in the time range T. The energy is normalized in the range of $[\underline{E}, \overline{E}]$, which are given by Eqs. (2) and (3). In this resource allocation, w and $(1-w)$ represent the relative importance of the energy consumption and the QoS, respectively. As w is larger, the optimal energy consumption decreases while the achieved fair QoS is lower. On the other hand, smaller w produces larger energy consumption and higher QoS. Equation (4) means that all tasks are executed with the same QoS, which implies the fairness. The total CPU utilization constraint is given by Eq. (5). Equation (6) represents the lifetime requirement, which means that the total energy consumption until the required lifetime L is smaller than the residual energy E.

3.2. Problem analysis

This subsection discusses how the optimal solution of Eqs. (1)–(8) is obtained. We denote a feasible solution $(x_1, \ldots, x_n, s)$ as (Q, s) since each x_i is uniquely determined for Q from $x_i = \phi_i(Q)$.

We assume that each ϕ_i is strictly convex and monotonically increasing. This assumption has been introduced in several resource allocation problems [4, 5]. A task to track a point in a video has such a resource consumption function for example.

Equation (1) is equivalent to Eq. (9) from Eq. (4):

$$\min\ W(Q,s) := \frac{wT\sum_{i=1}^{n}\phi_i(Q)\left(Kc_i^{\mathrm{CPU}}s^2 + e_i^{\mathrm{dev}}\right)}{\overline{E}-\underline{E}} - (1-w)Q. \tag{9}$$

Replacing Eq. (8) by Eq. (10), the problem is transformed into the relaxation problem Eqs. (2)–(6), (9), and (10). We firstly discuss the optimal solution (Q^{**}, s^{**}) of it.

$$s_1 \le s \le s_m. \tag{10}$$

From Eqs. (4)–(6), it is derived that s of any feasible solution (Q, s) is bounded by the functions $f(Q)$ and $g(Q)$:

$$s \ge f(Q) := \max\left(\frac{\sum_{i=1}^{n}\phi_i(Q)c_i^{\mathrm{CPU}}}{U-\sum_{i=1}^{n}\phi_i(Q)c_i^{\mathrm{dev}}}, s_1\right), \tag{11}$$

$$s \le g(Q) := \min\left(\left[\frac{E-T\sum_{i=1}^{n}\phi_i(Q)e_i^{\mathrm{dev}}}{TK\sum_{i=1}^{n}\phi_i(Q)c_i^{\mathrm{CPU}}}\right]^{\frac{1}{2}}, s_m\right). \tag{12}$$

As derived from Eqs. (11) and (12), the feasible region of Q is $Q \in [0, \min(1, Q^\star)]$, where $f(Q^\star) = g(Q^\star)$. Note that $f^{-1}(s)$ and $g^{-1}(s)$ exist on the domains $s \in (s_1, s_m]$ and $s \in [s_1, s_m)$, respectively, because both of the first terms in $f(Q)$ and $g(Q)$ are monotonic.

The following two propositions hold:

Proposition 1 *The optimal solution (Q^{**}, s^{**}) of Eqs. (2)–(6), (9), and (10) satisfies $s^{**} = f(Q^{**})$.*

proof Assume that (Q^{**}, s^{**}) satisfies $s^{**} > f(Q^{**})$. $W(Q^{**}, s^{**}) > (Q^{**}, f(Q^{**}))$ is derived from Eq. (9). It is a contradiction. □

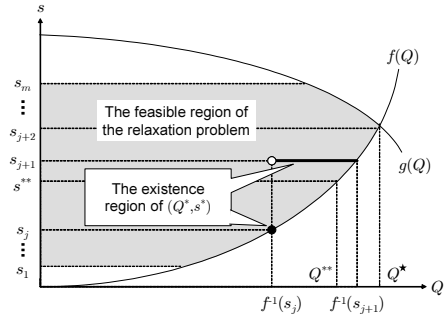
Proposition 2 *$W(Q, f(Q))$ is strictly convex in $Q \in [0, \min(1, Q^\star)]$.*

proof $W(Q, f(Q))$ is given by the following equation:

$$W(Q,f(Q)) = \frac{wKT}{\overline{E}-\underline{E}}\max\left[\frac{\left(\sum_{i=1}^{n}\phi_i(Q)c_i^{\mathrm{CPU}}\right)^3}{\left(U-\sum_{i=1}^{n}\phi_i(Q)c_i^{\mathrm{dev}}\right)^2}, \sum_{i=1}^{n}\phi_i(Q)c_i^{\mathrm{CPU}}s_1^2\right] + \frac{wT}{\overline{E}-\underline{E}}\sum_{i=1}^{n}\phi_i(Q)e_i^{\mathrm{dev}} - (1-w)Q. \tag{13}$$

Let $h(Q)$ be the function which is given

$$h(Q) = \frac{\sum_{i=1}^{n}\phi_i(Q)c_i^{\mathrm{CPU}}}{U-\sum_{i=1}^{n}\phi_i(Q)c_i^{\mathrm{dev}}}. \tag{14}$$



(Q^*, s^*): The optimal solution of Eqs. (1)–(8).
(Q^{**}, s^{**}): The optimal solution of Eqs. (2)–(6), (9), and (10).

Figure 2: The existence region of the optimal solution.

The strict convexity of ϕ_i guarantees that of $h(Q)$. By applying

$$\begin{aligned} f(Q) &= \max(h(Q), s_1), \\ \phi_i(\alpha Q_1 + (1-\alpha)Q_2) &< \alpha\phi_i(Q_1) + (1-\alpha)\phi_i(Q_2), \\ h(\alpha Q_1 + (1-\alpha)Q_2) &< \alpha h(Q_1) + (1-\alpha)h(Q_2), \end{aligned}$$

and $Q = \alpha Q_1 + (1-\alpha)Q_2$ ($\alpha \in [0, 1]$) to Eq. (13), we can derive the strict convexity of $W(Q, f(Q))$. □

These propositions implies that Eqs. (2)–(6), (9), and (10) are transformed into the optimization problem to find the minimum point of $W(Q, f(Q))$ for $Q \in [0, \min(1, Q^\star)]$. Since $W(Q, f(Q))$ is strictly convex, (Q^{**}, s^{**}) can be numerically calculated by an iterative method to find the minimal point of a convex function.

Now we discuss the optimal solution (Q^*, s^*) of Eqs. (1)–(8). If s^{**} is one of the available CPU speeds $\{s_1, \ldots, s_m\}$, (Q^*, s^*) equals to (Q^{**}, s^{**}). Otherwise, its existence region is given by Theorem 1. Figure 2 shows an illustration of Theorem 1.

Theorem 1 *Let (Q^{**}, s^{**}) be the optimal solution of Eqs. (2)–(6), (9), and (10), where $s_1 < \cdots < s_j < s^{**} < s_{j+1} < \cdots < s_m$. If the feasible region includes those with $s = s_j$ and $s = s_{j+1}$, the optimal solution (Q^*, s^*) of Eqs. (1)–(8) satisfies either of the following conditions:*

- *$s = s_j$ and $Q^* = f^{-1}(s_j)$.*
- *$g^{-1}(s_{j+1}) > f^{-1}(s_j)$, $s^* = s_{j+1}$, and Q^* is the minimum point of $W(Q, s_{j+1})$ in $Q \in \left(f^{-1}(s_j),\ \min(f^{-1}(s_{j+1}), g^{-1}(s_{j+1}))\right]$.*

proof The partial derivative of $W(Q, s)$ with respect to s is given by the following equation:

$$\frac{\partial W(Q,s)}{\partial s} = \frac{2wTk}{\overline{E}-\underline{E}}\sum_{i=1}^{n}\phi_i(Q)c_i^{\mathrm{CPU}}s > 0. \tag{15}$$

a. In the case that $(Q^*, s^*) \ne (f^{-1}(s_j), s_j)$ and $Q^* \le f^{-1}(s_j)$:
Equation (15) means that $W(Q, s)$ is monotonically increasing with respect to s. Therefore, s^* is the lower bound corresponding to Q^*, that is, $s^* = f(Q^*)$.

Since $W(Q, f(Q))$ is strictly convex from Proposition 2, it takes its minimum point at $Q = Q^{**}$ and monotonically decreases for $Q < Q^{**}$. Thus, $W(Q^*, s^*) = W(Q^*, f(Q^*)) > W(f^{-1}(s_j), s_j)$, which is a contradiction.

b. In the case that $f^{-1}(s_{j+1}) < Q^* \leq Q^\star$:
From the monotonicity of $W(Q, s)$ for s, $W(Q^*, s^*) \geq W(Q^*, f(Q^*))$. Since $s^{**} < s_{j+1}$ and $f^{-1}(\cdot)$ is monotonically increasing, $Q^{**} = f^{-1}(s^{**}) < f^{-1}(s_{j+1}) < Q^*$.

The strict convexity of $W(Q, f(Q))$ derives that it takes the minimum point at $Q = Q^{**}$ and monotonically increases for $Q > Q^{**}$. Thus, $W(f^{-1}(s_{j+1}), s_{j+1}) < W(Q^*, f(Q^*)) \leq W(Q^*, s^*)$. It is a contradiction.

c. In the case that $f^{-1}(s_j) < Q^* \leq f^{-1}(s_{j+1})$ and $s^* \neq s_{j+1}$:
Since $f(\cdot)$ is monotonically increasing, we can obtain $s_j < f(Q^*) \leq s^*$, which means $s^* > s_{j+1}$.

Since $W(Q, s)$ is monotonically increasing for s, $W(Q^*, s^*) > W(Q^*, s_{j+1})$. It is a contradiction.

It is proven from **a.–c.** □

3.3. Calculation of the optimal solution

From Theorem 1, the optimal solution (Q^*, s^*) of Eqs. (1)–(8) is obtained according to the following steps.

1. Calculate the optimal solution of (Q^{**}, s^{**}) of Eqs. (2)–(6), (9), and (10) and find (s_j, s_{j+1}) which satisfies $s_1 < \cdots < s_j < s^{**} < s_{j+1} < \cdots < s_m$.
2. Calculate $W(f^{-1}(s_j), s_j)$ by applying $Q = f^{-1}(s_j)$ and $s = s_j$ to Eq. (9).
3. Find the minimum point of $W(Q, s_{j+1})$ for $Q \in \left(f^{-1}(s_j), \min(f^{-1}(s_{j+1}), g^{-1}(s_{j+1}))\right]$.
4. (Q^*, s^*) is obtained by the comparison between the results of Steps 2 and 3.

Steps 1 and 3 can be performed by an iterative numerical method to find the minimum point of a convex function since $W(Q, f(Q))$ is strictly convex. For example, the Newton method can be applied. Steps 2 and 3 need calculation of $f^{-1}(s)$ and $g^{-1}(s)$. They are calculated by solving the following nonlinear equation forms:

$$0 = U - \sum_{i=1}^{n} \phi_i(Q) \left(\frac{c_i^{\mathrm{CPU}}}{s} + c_i^{\mathrm{dev}} \right), \tag{16}$$

$$0 = E - T \sum_{i=1}^{n} \phi_i(Q)(K c_i^{\mathrm{CPU}} s^2 + e_i^{\mathrm{dev}}). \tag{17}$$

From the convexity of ϕ_i, the solutions of these equation forms are obtained by a numerical method such as the Newton method. After Q^* is obtained, calculating $\phi_i(Q^*)$ provides the value of x_i^*. Thus, the optimal task frequency and CPU speed are calculated by a numerical method to find the minimum point of convex functions in Steps 1, 2, and 3.

4. Simulation

We conducted a simulation experiment to verify the effectiveness of our method. Shown in Fig. 3 is the optimal energy consumption and QoS of 5 tasks with respect to w. The trade-off is provided in $w \in [0.45, 0.63]$ because the optimal solution saturates in $w < 0.45$ and $w > 0.63$. Thus, by replacing w with $w' = (0.63 - 0.45)w + 0.45$ in this case, the trade-off may be provided effectively. The saturation range depends on the tasks' and CPU's characteristics. Thus, adapting w according to their characteristics may be required in the static allocator.

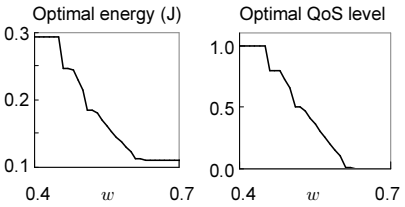


Figure 3: The optimal energy vs. QoS

5. Conclusion

Several battery-driven real-time systems have occasionally variant trade-off requirement between saving the energy consumption and guaranteeing the QoS. We formulated a static resource allocation problem to resolve the trade-off under the lifetime, energy, and fairness constraints. Also, from the convexity of resource consumption functions, we derived that the optimal allocation can be obtained by a numerical iterative method to solve some convex optimization problems.

Future work aims to reduce the convexity assumption of resource consumption functions.

Acknowledgement
This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Young Scientists (Start-up) (19800053).

References

[1] W. Yuan, K. Nahrstedt, S. Adve, D. Jones, and R. Kravets, "GRACE: Cross-layer adaptation for multimedia quality and battery energy," IEEE Transactions on Mobile Computings, vol.5, no.7, pp.799–815, July 2006.

[2] R.J. dos Santos Brito, Real-Time Scheduling of Tasks with Energy Constraints, Ph.D. thesis, Henri Poincaré University, June 2003.

[3] J. Liu, Real-Time Systems, Prentice Hall, 2000.

[4] D. Kang, S. Crago, and J. Suh, "A fast resource synthesis technique for energy-efficient real-time systems," In Proc. of IEEE Real-Time Systems Symposium, pp.225–234, Dec. 2002.

[5] R. Rajkumar, C. Lee, J. Lehoczky, and D. Siewiorek, "A resource allocation model for QoS management," In Proc. of IEEE Real-Time Systems Symposium, pp.298–307, Dec. 1997.