Optimizing QoS in Energy-Aware Real-Time Systems

Ríad Nassiffe
Department of Automation and Systems Engineering
Federal Univ. of Santa Catarina
riad@das.ufsc.br

Eduardo Camponogara
Department of Automation and Systems Engineering
Federal Univ. of Santa Catarina
camponog@das.ufsc.br

George Lima
Computer Science Department
Federal Univ. of Bahia
gmlima@ufba.br

We consider the problem of optimizing application quality of service subject to energy and real-time constraints. The system is composed of n sporadic and independent real-time tasks to be scheduled on a single CPU capable of operating at different frequency/voltage levels, which can be selected at run-time. Each task i may run at a selected frequency f_i and mode of operation k_i with each mode inducing a quality of service level. Due to schedulability and energy bounds, CPU frequencies and task modes should be selected by the system aiming at maximizing the overall system quality of service, an optimization problem formulated as:

$$P: \quad \max \quad \sum\nolimits_{i=1}^{n} \mathrm{QoS}(f_i, k_i) \tag{1a}$$

$$\text{s.t.}: \sum\nolimits_{i=1}^{n} u(f_i, k_i) \leq \mathrm{ub} \tag{1b}$$

$$\sum\nolimits_{i=1}^{n} e(f_i, k_i) \leq \mathrm{eb} \tag{1c}$$

$$f_i \in \mathcal{F}, k_i \in \mathcal{K}_i, \; i = 1, 2, \ldots, n \tag{1d}$$

where QoS represents the considered objective function, and constraints (1b) and (1c) are related to bounds on CPU (ub) and energy (eb), respectively. The set $\mathcal{F}$ may contain either the CPU frequency values allowed or represents a continuous frequency interval, a usual frequency model assumed in literature [1]. The set $\mathcal{K}_i$ represents a discrete set of modes that task i may operate. Different classes of optimization problems, with distinct degrees of difficulty, can be obtained depending on how $\mathcal{F}$ and $\mathcal{K}_i$ are modeled. In any case, the objective function QoS must capture some aspects: (I) *mode degradation* with priority given to the high quality operational modes; (II) *task value* which is associated with every task being strongly related to the application semantics; (III) *task weight* specifying the processing resources required by each task, thus a task that requires 10% of CPU should be treated differently from one that requires only 1%, possibly by receiving a higher task weight. Function (2) takes these aspects into account:

$$\mathrm{QoS}(f_i, k_i) = \frac{1}{|\mathcal{K}_i|} \left(k_i - 1 + w_i u_i(f_i, k_i)\right) \tag{2}$$

where $\mathcal{K}_i = \{1, 2, \ldots\}$ has the indices of the task modes and $w_i \in [0, 1]$ is the task value. Mode k_i has quality higher than mode k_i' whenever $k_i > k_i'$. Notice that QoS will be a value between 0 and 1 since the CPU utilization $u_i(f_i, k_i)$ cannot exceed the unit.

The reduction of CPU frequency results in an increase of task execution time which can further delay the completion of the tasks, specially those that need more resources than predicted, a common scenario for soft real-time systems. For this reason the task values should be adjusted depending on the CPU frequency, with higher values for tasks running at high frequency. Assuming that $w_i = f_i^2$, $\mathcal{F} = \{100\%, 50\%, 25\%\}$ and a task with 3 modes, where the utilization at $f_i = 100\%$ are respectively $0.042, 0.34, 0.60$, the QoS function (2) ensures that a configuration with high mode and frequency will always have greater value/system benefit, thereby avoiding unnecessary degradation of the system as shown in Table 1.

Table 1: Task configuration benefits

	$QoS(f_i, k_i)$		
	$k_i = 1$	$k_i = 2$	$k_i = 3$
$QoS(100\%, k_i)$	0.014	0.448	0.866
$QoS(50\%, k_i)$	0.007	0.391	0.766
$QoS(25\%, k_i)$	0.003	0.362	0.716

Scheduling problems can be modeled as continuous or discrete as shown in [3]. Continuous and convex models can be efficiently solved by interior-point methods [2]. Discrete models can be solved with relaxation-based heuristics [3, 4] or with specialized algorithms. Applying such techniques to problem (1a)-(1d) has led to promising preliminary results.

1. REFERENCES

[1] H. Aydin, R. Melhem, D. Mossé, and P. Mejía-Alvarez. Power-Aware Scheduling for Periodic Real-Time Tasks. *IEEE Transactions on Computers*, 53(5):584–600, 2004.

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[3] E. Camponogara, A. B. de Oliveira, and G. Lima. Optimization-Based Dynamic Reconfiguration of Real-Time Schedulers with Support for Stochastic Processor Consumption. *IEEE Transactions on Industrial Informatics*, 6(4):594–609, Nov 2010.

[4] R. Nassiffe, E. Camponogara, and G. Lima. Optimizing Quality of Service in Real-Time Systems Under Energy Constraints. *ACM OSR*, 46(1):82–92, 2012.