Utility aware Energy Constrained Real-Time Scheduling

Michael Shell, *Member, IEEE,* John Doe, *Fellow, OSA,* and Jane Doe, *Life Fellow, IEEE*

***Abstract*—Mapping and Scheduling of real-time tasks and messages of conditional DAGs in automotive systems.**

***Index Terms*—Real-Time Systems, Automotive Systems, Conditional DAG.**

I. System Model and Problem Formulation

In this work, we use the notation (for example) $\sum_j$ to mean $\sum_{j\in[1,N_i^{job}]}$.

Problem Mapping and Scheduling of segmented real-time tasks. Non-preemptive. Energy budget. Maximize weighted average utility.

Processor Model. Given a set $\mathcal{P}$ of N^{prc} heterogeneous processors such that each processor P_x executes with a discrete set of N^{frq} frequency $\mathcal{F}_x = \{f_{x,z}\}$ such that $f_{x,z}$ is normalized with respect to the $f^{max} = \max_x\{f_{x,N^{frq}}\} = 1$ in the system. Furthermore, frequency values in $\mathcal{F}_x$ are assumed to be arranged in an increasing order.

Task Model. Given a set $\mathcal{T} = \{T_i\}$ of N^{tsk} real-time periodic tasks. Each task T_i is specified with the following parameters.

- Each task is activated periodically with a period of p_i, each such activation is referred to as the *job* of this task.
- j^{th} job of T_i is activated at $(j-1)p_i$ with a *deadline* jp_i.
- Each task T_i (and so does each job of this task) has a *mandatory* segment and N_i^{seg} optional segments. While the mandatory segments of all tasks must finish by the respective deadline, executing optional segment allows refining the quality of output, thus higher *utility*. Evidently, executing optional segment of a task is optional but if executed provides higher utility. Let $O_{i,k}$ be the k^{th} optional segment of T_i. Executing $O_{i,k}$ indicates all previous optional segment of this task are also executed. Let e_i be the time to execute mandatory segment of task T_i with the highest frequency f^{max}. Whereas $e^O_{i,k}$ be the execution time of k^{th} optional segment of T_i with f^{max}.

Let $h = \text{lcm}\{p_i\}$ be the hyper-period. Then, the number of jobs of T_i in the hyper-period is $N_i^{job} = \frac{h}{p_i}$. We allow jobs of a task to execute on a different processor with a different frequency. However, once a job starts executing it cannot be preempted (that is to say, scheduling is **non-preemptive**), it cannot migrate to another process or execute with a different frequency [1].

Utility Model. We assume that the utility gained by executing optional segment is weighted, however remains the same for all jobs of a task. Furthermore, we assume that utility values are already adjusted and u_i denote the utility gained by per unit execution of optional segment of task T_i. In this work, we consider a *linear* utility function.

Energy Model. Following the common convention, we follow the following energy model. The energy consumption to execute $T_{i,j}$ on processor P_x with z^{th} frequency while $O_{i,k}$ is the last optional segment to be executed.

$$E_{i,j,k,x,z} = \alpha \frac{e_i + \sum\limits_{q=1}^{k} e^O_{i,q}}{f_{x,z}} \left(\beta f_{x,z}^2 + e_i + \sum_{q=1}^{k} e^O_{i,q}\right). \quad (1)$$

In this work, the total energy consumption should not exceed energy budget B.

Problem Statement. The problem of utility aware scheduling of real-time task with energy budget is referred to USRT. USRT problem is to compute a mapping $T_{i,j} \mapsto \langle P_x, f_{x,z}, k, t^{beg}_{i,j}\rangle$ to optimize average weighted utility such that following constraints are satisfied.

1) timing constraint is satisfied for each job in the hyper-period that is to say, the mandatory segment of all jobs and selected optional segment must be executed before respective deadline
2) total energy consumption does not exceed the energy budget B
3) scheduling constraints are satisfied - non-preemption, no migration, no change in frequency, exclusivity (only one job can be executed by a processor at a time)

II. ILP Formulation

Let $V(ijkxyz)$ be a binary (decision) variable such that $V(ijkxyz) = 1$ if $T_{i,j}$ is mapped to processor P_x and executes with the frequency $f_{x,z}$ up to k^{th} segment at the y^{th} position. Note, there can be up to $\sum\limits_{i=1}^{N^{tsk}} N_i^{job}$ jobs could be mapped (scheduled in a non-preemptive fashion) on a processor and thus, $y \in [1, L]$. Then, USRT is to *compute* $V(ijkxyz)$ such that following constraints are satisfied.

[C1.] Each job is executed exactly once. Only one processor, one position, one frequency, and up to zero or more optional segments.

$$\sum_{k,x,y,z} V(ijkxyz) = 1, \forall i, j \quad (2)$$

[C2.] At any position on a processor, at most one job can be executed.

$$\sum_{i,j,k,z} V(ijkxyz) \le 1, \forall x, y \quad (3)$$

[C3.] Timing constraint is satisfied for each allocation of a processor.

$$eft(x,y) \leq deadline(x,y), \forall x,y \tag{4}$$

where $eft(x,y)$ and $deadline(x,y)$ are computed as described in the following. The number of allocations A_x on the processor P_x is given as

$$A_x = \sum_{i,j,k,y,z} V(ijkxyz) \tag{5}$$

Then, earliest execution start time of the first allocation on P_x is the release time of the corresponding job given as

$$est(x,1) = \sum_{i,j,k,z} V(ijkxyz) \times (j-1) \times p_i \tag{6}$$

and the earliest finish time is

$$eft(x,1) = est(x,1) + \sum_{i,j} V(ijkxyz) \times \frac{e_i + \sum_{q=1}^{k} e^O_{i,q}}{f_{x,z}} \tag{7}$$

Likewise, the earliest execution start time of the $l^{th}(l \geq 2)$ allocation on P_x is

$$est(x,l) = \max \left\{ \sum_{i,j,k,z} V(ijkxyz) \times (j-1) \times p_i, eft(x,l-1) \right\} \tag{8}$$

and the earliest finish time is

$$eft(x,l) = est(x,l) + \sum_{i,j} V(ijkxyz) \times \frac{e_i + \sum_{q=1}^{k} e^O_{i,q}}{f_{x,z}} \tag{9}$$

The deadline is

$$deadline(x,l) = \sum_{i,j,k,z} V(ijkxyz) \times j \times p_i \tag{10}$$

Need to linearize maximum computation.

[C4.] Total energy consumption E^{total} does not exceed energy budget.

$$E^{total} \leq B \tag{11}$$

Objective: The objective is optimize (maximize) sum of average weighted utility in the hyper-period accumulated by executing optional segments.

- We assume utility values are already adjusted to incorporate the weight.
- Let u_i denote the utility gained by per unit execution of optional segment of task T_i.

Utility gained by executing job $T_{i,j}$ is

$$U^{job}_{i,j} = u_i \times \sum_{k,x,y,z} V(ijkxyz) \times e^O_{i,q} \tag{12}$$

Utility gained by task T_i over all jobs in the hyper-period

$$U^{tsk}_i = \sum_j U^{job}_{i,j} \tag{13}$$

The objective Φ is

$$\Phi = \frac{\sum_i U^{tsk}_i}{N^{tsk}} \tag{14}$$

III. Scratchpad

- Preemptive
- non-linear utility function
- energy optimization

References

[1] N. Kumar, J. Mayank, and A. Mondal, “Reliability aware energy optimized scheduling of non-preemptive periodic real-time tasks on heterogeneous multiprocessor system,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 4, pp. 871–885, 2019.