Approximation Algorithms for Power Minimization of Earliest Deadline First and Rate Monotonic Schedules[*]

Sushu Zhang, Karam S. Chatha and Goran Konjevod
Department of Computer Science and Engineering, Arizona State University, Tempe, AZ, 85287.
sushu.zhang@asu.edu, kchatha@asu.edu, goran@asu.edu

ABSTRACT

We address power minimization of earliest deadline first and rate-monotonic schedules by voltage and frequency scaling. We prove that the problems are NP-hard, and present $(1+\epsilon)$ fully polynomial time approximation techniques that generate solutions which are guaranteed to be within a specified quality bound (QB$=\epsilon$) (say within 1% of the optimal). We demonstrate that our techniques can match optimal solutions when QB is set at 1%, out perform existing approaches [1] even when QB is set at 10%, generate solutions that are quite close to optimal ($< 5\%$) even when QB is set at higher values (25%), and execute in a fraction of a second (with QB $> 5\%$) for large 100 node task sets.

Categories and Subject Descriptors

D.4.7 [**Operating Systems**]: Organization and Design—*Real time systems and embedded systems*

General Terms

Algorithms, Performance

Keywords

Low power design, Earliest deadline first, Rate monotonic

1. INTRODUCTION

Embedded processors have exploited the increase in operating frequencies due to technology scaling to address the increasing performance requirements of applications. However, in the nanoscale era, benefits of technology scaling are contingent upon overcoming the power consumption challenges. Technology scaling has enabled higher transistor counts, and also contributed to an increase in the static power consumption. These two factors have resulted in a sharp increase in both power consumption and power densities of embedded processors. Power consumption is of direct consequence to a large number of portable embedded system applications that are constrained by battery lifetimes. Further, even tethered applications such as set top boxes, edge and back bone routers have thermal budgets which in turn translate into power consumption constraints.

Dynamic voltage frequency scaling (DVFS) [1] and dynamic power management (DPM) [2] have emerged as two primary system-level optimizations for low power design on embedded processors. DVFS exploits the CMOS property that a linear reduction in the supply voltage results in a cubic reduction in the power consumption at the expense of a linear slow down in the processor frequency. DVFS schemes exploit this relationship to provide variable operating voltages and corresponding frequencies for the processor. DPM exploits the idle times and turns off the power supply to several subsystems of the processing element. Therefore, DPM reduces the stand-by power or leakage power consumption of the application. It is preferable to derive maximum benefit of the DVFS policies (which can generate cubic reduction in active power consumption) and then exploit DPM [1].

Earliest deadline first (EDF) and rate monotonic (RM) are the two main real time scheduling algorithms for periodic task sets on uni-processor architectures [3]. In both the schemes a task can be released at anytime within its period, and it must finish execution by the end of its period. In other words the deadline of each task is equal to its period. EDF is a dynamic priority scheme that assigns the highest priority to the task with the earliest deadline. RM is a static priority scheme that assigns the highest priority to the task with shortest period. Consequently, for large task sets EDF has a higher utilization bound of 1 as opposed to RM whose utilization bound is given by 0.69 ($\ln 2$).

The paper addresses the system-level low power design of set of tasks to be executed under EDF or RM scheduling scheme on an embedded processor that supports DVFS. Each task is specified by its period and known execution times and power consumption at the various voltage/frequency states of the target processor. The objective is to assign a voltage and frequency state for execution of each task such that the total power consumption of the application is minimized subject to the processor utilization bounds of EDF and RM scheduling schemes.

It can be shown that the two problems as specified above are NP hard. We present fully polynomial time approximation schemes (FPTAS) for the problems. The FPTAS can be utilized to generate solutions that are guaranteed to be within a designer specified approximation bound (e.g. within 1% of the optimal power consumption) in polynomial time. We present experimental results that evaluate the proposed techniques with both real and synthetic applications, and comparisons with optimal and existing [4] approaches.

[*]The research presented in this paper was supported by grants from the National Science Foundation (Career CCF-0546462 and CNS-0509540), and the Consortium for Embedded Systems.

The paper is organized as follows: Section 2 discusses the previous work, Section 3 defines the problem, Section 4 presents the fully polynomial time approximation scheme for the problem , Section 5 discusses the experimental results, and finally Section 6 concludes the paper.

2. PREVIOUS WORK

Jha et al. [1] and Benini et al. [2] give a survey of the existing DVFS and DPM techniques, respectively. There exists a significant body of research on efficient algorithms for DVFS in hard real-time systems. These can be classified on the basis of the following categories: i) offline [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 4, 16, 17, 18] versus online [19, 20, 21, 22, 23] schemes for voltage/frequency state assignment, ii) inter-task DVFS [5, 7, 8, 9, 10, 11, 12, 14, 15, 4, 16, 17, 19, 20, 22, 23, 18] versus intra-task [6, 13, 21] approaches, and iii) continuous voltage/frequency scaling [5, 7, 9, 11, 19] versus discrete active states [6, 8, 10, 12, 13, 14, 15, 4, 16, 17, 20, 21, 22, 23].

Yao et al. [5] proposed an optimal offline low power scheduling algorithm that assumed a processor with continuous voltage/ frequency scaling. However, realistic embedded processors only offer discrete voltage/frequency states. Ishihara et al. [6] proposed an optimal low power offline scheduling algorithm where every task is executed with at most two discrete voltage states. Although intra-task DVFS approach can possibly result in greater power consumption savings than inter-task DVFS technique, it is not easy to implement in the operating system and requires a higher overhead. These drawbacks have also been specifically recognized by others [24], and are substantiated by the much larger body of work on inter-task optimizations versus intra-task approaches. Due to the above mentioned observations in the following discussions we primarily focus on online and offline low power scheduling algorithms for inter-task DVFS with discrete active states.

Pillai et al. [8] and Jejurikar et al. presented [12] offline and online heuristic schemes for integration of DVFS with real time schedulers. The technique presented by Jejurikar et al. also considered the leakage power consumption of the peripheral components that are present in the system. Yan et al. [10] in addition to the traditional DVFS also considered adaptive body biasing (ABB) to minimize the leakage power consumption of an embedded processor. Mochocki et al. presented heuristic algorithms for offline [14] and online [23] DVFS that considered switching overheads between voltage states. Xie et al. [15] presented an exponential time exact algorithm based on branch and bound, and a linear time heuristic for DVFS that considered switching overheads and power consumption of peripheral components. Mejia-Alvarez et al. [4] and Yang et al. [18] proposed a greedy heuristics based on modelling the low power scheduling problem as a variations of the standard knapsack problem. All the above mentioned approaches either propose heuristic algorithms or exponential run time exact approaches for DVFS. In contrast we propose polynomial time algorithms for offline DVFS that generate solutions for EDF and RM schedulers which are guaranteed to be within a designer specified bound from the optimal. Chen et al. [16] and Zhong et al. [17] presented fully polynomial time and pseudo polynomial time approximation algorithms for the low power DVFS problem, respectively. Our approach is a fully polynomial time approximation scheme which has a lower complexity than either of these two approaches.

3. PROBLEM DEFINITION

Given:

- n independent periodic tasks $\mathcal{X}\{x_1, \dots, x_n\}$ specified in the ascending order of their period d_i,
- a target embedded processor architecture with multiple active voltage and corresponding frequency states[1] $\mathcal{V}\{v_1, \dots, v_m\}$,
- for each task $x_i \in \mathcal{X}$ and each active state $v_j \in \mathcal{V}$, p_{ij} and t_{ij} that denote the power consumption and execution time of the task, respectively,

The objective is to minimize the power consumption when the tasks are executed by EDF (or RM) scheduling schemes subject to:

- each task is executed at a unique voltage state of the processor,
- every task is finished before its next request, and
- the utilization bound of valid EDF (or RM) scheduling is satisfied.

In the following sections, we first prove that the problem as specified above is NP hard and then derive an approximation algorithm for the same. In the following discussion we address the problem in the context of EDF scheduling and then present modifications for RM scheduling. The switching overhead between frequency states or tasks is assumed to be negligible. We denote the two problems as LP-EDF and LP-RM.

4. APPROXIMATION ALGORITHMS

LP-EDF and LP-RM problems can be proved to be NP-hard. We derive fully polynomial time approximation schemes (FPTAS) for the problems. Given an NP-hard minimization problem Π with an objective function f_Π, an algorithm $\mathcal{A}$ is an approximation scheme for Π if given an instance I of the problem, and an error parameter ϵ it outputs a solution s such that $f_\Pi(I,s) \le (1+\epsilon)\cdot OPT$ where OPT is the optimal solution. $\mathcal{A}$ is a FPTAS if its running time is bounded by a polynomial in the size of the instance I and $1/\epsilon$. FPTAS is the best one can hope for a problem that is NP-hard [25].

We consider scheduling the tasks over the hyper-period D which is the least common multiple of $d_1, d_2, \dots, d_n$. For a task x_i there are $\frac{D}{d_i}$ instances to be executed in D. Let e_{ij} be the total energy consumption of the task instances when it is executed at voltage v_j, thus $e_{ij} = p_{ij} \times t_{ij} \times \frac{D}{d_i}$. Let a_{ij} denote a $0/1$ variable that is '1' if task x_i is assigned to execute at voltage/frequency state v_j (otherwise $a_{ij} = 0$). The power consumption of the set of tasks is given by:

$$P = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} \cdot e_{ij}}{D}$$

The numerator represents the total energy consumption (E) due to the execution of the tasks at their assigned voltage/frequency states in the hyper-period D. The voltage/frequency assignment problem for low power EDF (LP-EDF) or RM (LP-RM) schedules can be stated as:

$$\min(E) \text{ where } E = \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} \cdot e_{ij}$$

such that

$$\sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} \frac{t_{ij}}{d_i} \le U_{EDF/RM} \quad (1)$$

$$\forall i \sum_{j=1}^{m} a_{ij} = 1 \quad (2)$$

[1]We assume without loss of generality that at a particular voltage the processor operates at a unique frequency. The proposed techniques can also address the more general case of multiple operating frequencies at a particular processor voltage

In the formulation Equation 1 denotes that the sufficient condition on the utilization bound of the EDF or RM schedule is satisfied.

THEOREM 1. *The LP-EDF and LP-RM problems as stated above are NP-hard.*

PROOF. The problems can be shown to be NP-hard by a reduction from the multiple-choice knapsack problem (MCKP) [26][4]. The energy minimization objective is replaced by the goal of maximizing energy savings. Let $E_{\max}$ be the maximum energy consumption of any task, that is $E_{\max} = \max(e_{ij}), \forall x_i \in \mathcal{X}, v_j \in \mathcal{V}$. The energy savings due to a task x_i operating at voltage/frequency state v_j is given by the difference between the $E_{\max}$ and e_{ij}. Finding an optimal solution to the ILP formulation with the energy savings maximization objective is equivalent to solving MCKP, which is NP-hard [26]. □

There are known FPTAS for solving the MCKP problem. Chandra et al. [27] present the first FPTAS for MCKP problem. Lawler et al. [28] and Kellerer et al. [26] proposed similar FPTAS with running time much better than that of the scheme in [27]. However, as is often the case, equivalence of finding optimal solutions to these two problems does not imply that an approximation algorithm for MCKP can be directly used with the same approximation guarantee for LP-EDF and LP-RM. In fact, one can easily prove that FPTAS solutions to MCKP in some cases translate into poor solutions to LP-EDF and LP-RM. Thus even though there exist polynomial-time approximation schemes for MCKP, finding good approximation algorithms for LP-EDF and LP-RM are open problems. We first give an exact pseudo-polynomial time algorithm for LP-EDF and LP-RM based on dynamic programming, and then use it to construct a FPTAS.

4.1 Optimal algorithms

The exact algorithms are based on a dynamic programming algorithm for the knapsack problem [25] that runs in pseudo-polynomial time. Given $E_{\max}$ as defined above, $nE_{\max}$ is an upper bound on the energy consumption of any solution. Let $S_{i,E}$ denote an assignment of the i tasks $x_1, \ldots, x_i$ to voltages such that their energy consumption is at most E and the total processor utilization due to these i tasks is minimized. Let $U(i,E)$ be this minimum processor utilization. If $S_{i,E}$ does not exist, define $U(i,E) = \infty$. $U(1,E)$ is known for $E \in [1, \ldots, nE_{\max}]$. The recurrence relation for the dynamic programming algorithm is given by:

$$U(i,E) = \min_{j \in [1,m]} (U(i-1, E - e_{ij}) + \frac{t_{ij}}{d_i}).$$

From this recurrence we can find $U(n,E)$ for all $E \in [1, nE_{\max}]$. The optimum solution is then S_{n,E^*}, where

$$E^* = \{\min E | U(n,E) \leq U_{EDF/RM}\}.$$

The recurrence leads to an algorithm that loops over tasks $i \leq n$, energy values $e \leq nE_{\max}$ and m voltage states to construct a two-dimensional table indexed by tasks and energy values, so that entry (i,e) contains $U(i,e)$. The table is constructed in order, so that before considering (i,e), the first $i-1$ rows are filled in. For each (i,e), we compute $U(i,E)$ by looping over different voltages, as indicated in the recurrence above.

The computational complexity of the algorithm is $O(n^2 m E_{\max})$, because there are $n(nE_{\max})$ entries in the table, and determining each requires m steps.

4.2 $(1+\epsilon)$-FPTAS

The exact algorithms described above are not polynomial because their running time includes a factor $E_{\max}$, which could be exponential in the number of tasks and voltage states. We next describe algorithms parameterized by δ. The approximation guarantee for these algorithms is $(1+2\delta)$. However, these are still FPTAS as we can get a $(1+\epsilon)$ approximation by invoking the algorithms with parameter $\delta = \epsilon/2$. To get polynomial algorithms with approximation guarantee $(1+2\delta)$, we first show that if the optimal energy consumption E^* was given as part of the input, we could adapt the knapsack polynomial-time approximation scheme [25] and get a $(1+2\delta)$-approximate solutions to our problems. In fact, we show the following:

LEMMA 1. *Let $E_{ub} \leq \alpha E^*$ for some $\alpha \geq 1$. If $probe(E_{ub})$ succeeds (where probe is the function defined in lower half of Figure 1), then the solution found by the call to the dynamic programming procedure consumes at most $(1+\alpha\delta)E^*$ energy.*

PROOF. Given δ and the upper bound E_{ub}, $probe$ first scales the energy consumption values: let $K = \delta E_{ub}/n$ and replace e_{ij} by $e'_{ij} = \lceil e_{ij}/K \rceil$ for every i and j. The next step is to invoke the exact dynamic programming algorithm described in the previous section using the modified energy consumption values e'_{ij}. $U'(n,E')$ is identical to $U(n,E)$ except that it operates on scaled values. The program returns an optimal solution A to the scaled instance of the problem. Let A^* denote the optimal solution to the original instance. Let $E'(A)$ denote the cost of A in terms of the modified energy consumption values e'_{ij}. To simplify the notation, we use $ij \in A$ to denote that in the solution A, task x_i is assigned voltage v_j, thus contributing e_{ij} to the energy consumption (or e'_{ij} in the scaled instance). We have

$$\begin{aligned} E(A) \leq KE'(A) &= \sum_{ij \in A} Ke'_{ij} \leq \sum_{ij \in A^*} Ke'_{ij} \\ &\leq \sum_{ij \in A^*} (e_{ij} + K) \leq E^* + nK = E^* + \delta E_{ub} \\ &\leq E^*(1+\alpha\delta). \end{aligned}$$

The first step is true because of rounding after scaling. The second step expands the energy consumption of A term by term. The third step is true because A is the optimal solution for the scaled instance, and so in terms of e' it is cheaper than A^*. The fourth step follows because by the definition of e'_{ij}, we have $e_{ij} \leq Ke'_{ij} \leq e_{ij} + K$. The remaining steps follow from the definitions of K and α. □

We use the function $probe$ as a building block for our algorithms. The full algorithms are shown in Figure 1. It consists of a binary search on the sequence $(1, 2, 2^2, \ldots, 2^i, \ldots, 2^N)$, where $N = \lceil \lg nE_{\max} \rceil$.

LEMMA 2. *If $probe(E)$ returns failure, then $E^* > E$.*

PROOF. Suppose $E \geq E^*$. Then by the definition of scaling, the optimal solution A^* to the original instance has scaled energy consumption at most

$$E'(A^*) \leq \left\lceil \frac{E^*}{K} \right\rceil + n \leq \left\lceil \frac{E}{K} \right\rceil + n.$$

Since $probe(E)$ invokes the dynamic program with the upper bound $\lceil \frac{E}{K} \rceil + n$, there exists a feasible solution and $probe(E)$ will succeed. □

THEOREM 2. *The approximation ratio of LP-EDF/LP-RM FPTAS is $(1+2\delta)$.*

PROOF. Let 2^h be the value returned by the binary search. Let $E(A)$ be the energy consumption of the solution found by $probe(2^h)$.

```
LP-EDF/LP-RM FPTAS($\delta$):
Initial: $N = \lceil \lg nE_{\max} \rceil$, $l = 1$, $r = N$;
while ($l < r$)
{
    $h = \lfloor \frac{l+r}{2} \rfloor$.
    if ($probe(2^h)$ = $success$) then $r = h$;
    else $l = h$;
}
$h = r$;
return $2^h$;
```

```
probe($E$)
{
    $K = \frac{\delta E}{n}$;
    $e'_{ij} = \lceil \frac{e_{ij}}{K} \rceil$, $E' = \lceil \frac{E}{K} \rceil + n$;
    if ($U'(n, E') \leq U_{EDF/RM}$)) {return $success$;}
    else {return $failure$;} endif;
}
```

Figure 1: LP-EDF and LP-RM: FPTAS

We have the following two cases:
CASE I: $2^h \leq E^*$. Then by Lemma 1, $E(A) \leq (1+\delta)E^*$.
CASE II: $2^h > E^*$. As 2^h is the smallest value for which the probe succeeds we have $2^h < 2E^*$. Now Lemma 1 implies $E(A) \leq (1+2\delta)E^*$. □

LEMMA 3. *The running time of LP-EDF/LP-RM FPTAS is bounded by* $O(\frac{n^2 m}{\delta} \lg\lg(nE_{\max}))$.

PROOF. The binary search is applied to the N-element sequence $(1, 2, \ldots, 2^i, \ldots, 2^N)$, where $N = \lceil \lg(nE_{\max}) \rceil$. Therefore, $probe$ is invoked at most $O(lg(N)) = O(\lg\lg(nE_{\max}))$ times. Each call to $probe$ requires $O(\frac{n^2 m}{\delta})$ time. Thus the overall running time of the algorithm is $O(\frac{n^2 m}{\delta} \lg\lg(nE_{\max}))$. While this expression contains the term $E_{\max}$, the double logarithm ensures that the running time is not only polynomial in the size of the input, but also that the extra term $\lg\lg(nE_{\max})$ is only logarithmic in the input size. □

THEOREM 3. *LP-EDF/LP-RM FPTAS are fully polynomial approximation schemes for LP-EDF and LP-RM problems, respectively.*

PROOF. From Theorem 2 and Lemma 3, the proof follows. □

5. RESULTS

We present and analyze the results of experimentation that was performed to evaluate our techniques. We evaluated both LP-EDF and LP-RM FPTAS for multimedia benchmark applications and synthetic task sets. In both the experiments we compared our techniques against a non-DVFS approach, optimal designs, and SGA and EGA algorithms proposed by Mejia-Alvarez et al. [4]. The non-DVFS approach executes the tasks at their highest voltage/ frequency state. The optimal designs were obtained by utilizing the exact algorithm discussed in Section 4.1. The Intel StrongARM 1100 processor was considered as the target embedded processor for the two experiments. The optimization techniques were coded in C++ and the experimentations were performed on a Pentium M/1.6GHz/512MB WindowsXP PC.

5.1 Results for multimedia benchmarks

We considered four applications drawn from the multimedia domain namely JPEG decoding, MPEG2 decoding, MP3 encoding

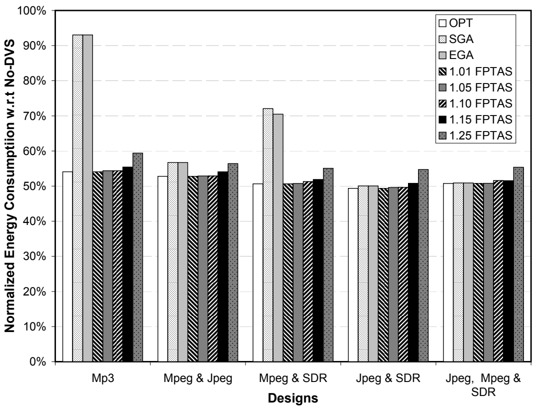

Figure 2: LP-EDF FPTAS: Multimedia benchmarks

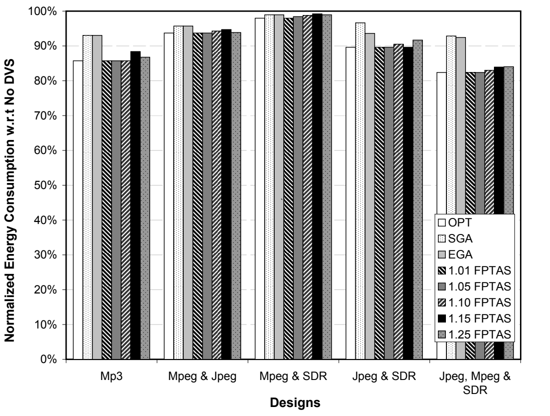

Figure 3: LP-RM FPTAS: Multimedia benchmarks

and software defined radio (SDR). The JPEG decoding algorithm was modeled by four tasks consisting of: variable length decoding, huffman decoding, inverse-zigzag and quantization, and IDCT. An MPEG2 stream consists of I, P and B pictures. Decoding an I picture consists of the following tasks: preprocessing, variable length decoding, inverse zigzag and de-quantization, and IDCT. P and B picture decoding consist of preprocessing and motion compensation. The MP3 encoding algorithm was modeled by three tasks consisting of pulse code modulation, filtering, and huffman encoding. Finally, the SDR application was obtained from Niyogi et al. [29] and consisted of low pass filter, demodulator and equalizer.

The Intel StrongARM 1100 processor was run at the following specifications: 1.5 V - 206 MHz, 1.4 V - 192 MHz, 1.2 V - 162 MHz and 1.1 V - 133 MHz. We obtained execution times (CPU run time) and average power consumption estimates of the tasks in the multimedia applications by utilizing the JouleTrack simulator [30] for the StrongArm processor. We considered the design of application sets with the period constraints for all tasks in a particular application specified as follows: JPEG = $1ms$, MPEG2 = $900\mu s$, MP3 = $45ms$, and SDR = $8ms$. For the integrated JPEG, MPEG2 and MP3 design the periods were specified as $1.5ms$, $1.5ms$ and $12ms$, respectively. We implemented the designs with both LP-EDF FPTAS and LP-RM FPTAS and with quality bounds of 1% ($\epsilon = 0.01$), 5% ($\epsilon = 0.05$), 10% ($\epsilon = 0.10$), 15% ($\epsilon = 0.15$) and

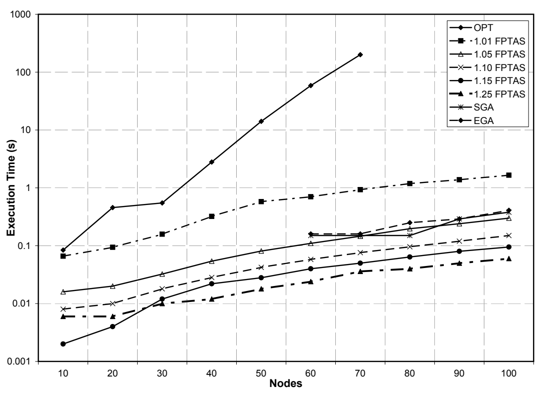

Figure 4: LP-EDF FPTAS: execution time

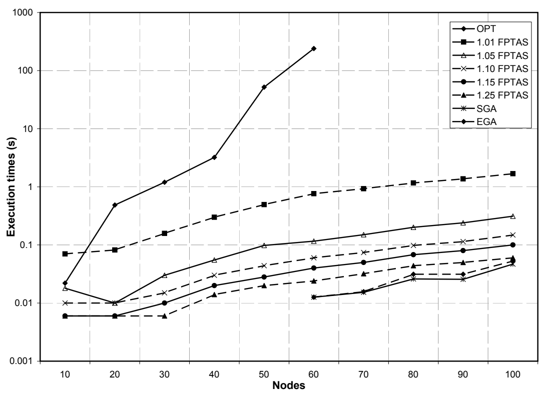

Figure 5: LP-RM FPTAS: execution time

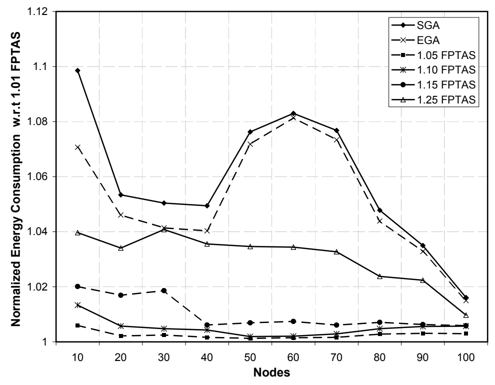

Figure 6: LP-EDF FPTAS: actual design quality

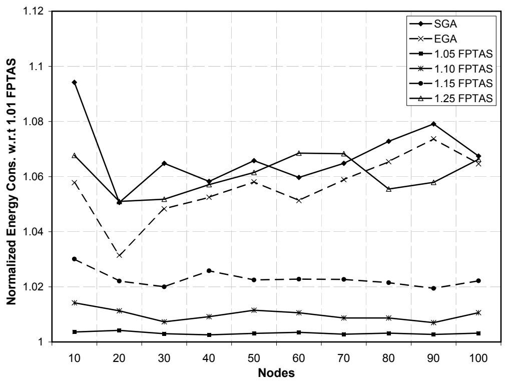

Figure 7: LP-RM FPTAS: actual design quality

25% ($\epsilon = 0.25$). The results are plotted in Figure 2 and 3 for EDF and RM schedulers, respectively. The plots depict the normalized energy reduction due to a particular approach in comparison to no DVFS technique.

Evaluation of LP-EDF FPTAS Both SGA and EGA give inferior results in comparison to the optimal in all cases. In fact on an average the SGA and EGA are over 1.25 (max = 1.72) and 1.24 (max = 1.72) times the optimal, respectively. In contrast LP-EDF FPTAS with a bound of 1 % is able to match the optimal solution in all cases. Even with a quality bound of 25 % LP-EDF FPTAS is able to out perform SGA and EGA in 3 out of the 5 cases, and is on an average within 1.09 (max = 1.11) of the optimal.

Evaluation of LP-RM FPTAS As the RM scheduler is constrained by much lower utilization bound, the energy consumption is higher in comparison to the EDF scheduler. Similar to the EDF scheduler, both SGA and EGA give inferior results in comparison to the optimal in all cases. On an average the SGA and EGA are 1.06 (max = 1.13) and 1.06 (max = 1.12) times the optimal, respectively. Again, LP-RM FPTAS with a quality bound of 1 % is able to match the optimal solution in all cases. LP-RM FPTAS with a quality bound of 25 % out performs SGA and EGA in all cases, and is on an average within 1.01 (max = 1.02) of the optimal.

Summary We can conclude that for realistic applications both LP-EDF and LP-RM FPTAS are able to match the optimal with a quality bound of 1 %, out perform SGA and EGA in most instances with a quality bound of 25 % and also generate high quality solutions with a quality bound of 25 %.

5.2 Results for synthetic task sets

We evaluated the proposed techniques by experimenting with large synthetic task sets with up to 100 nodes. The number of tasks in each set were varied from 10 to 100 in steps of 10 with 10 task sets at each value. Each task was assumed to run on the StrongArm processor with 5 voltages (0.90V, 1.00V, 1.20V, 1.30V, 1.50V). The workload of the tasks was varied uniform randomly from 10^2 to 10^8 clock cycles. In the case of EDF scheduler, for each task set, 10% of the tasks were set to be high utilization tasks at lowest operating frequency. As the RM schedule has lower utilization than EDF scheduler, only 5% of total tasks were assigned as high utilization. The utilization of tasks with high utilization was varied uniform randomly from $\frac{1}{N}$ to $\frac{1.5}{N}$ where N is the number of jobs in the task set. The utilization of low utilization tasks was varied uniform randomly from 0.0001 to $\frac{1}{N}$. We generated designs by executing the optimal algorithm, SGA, EGA and both FPTAS-EDF and FPTAS-RM. We set the quality bounds of our algorithm at 1%, 5%, 10%, 15% and 25%. We recorded the execution times of the various techniques and also compared the actual quality of results with the solution of 1% FPTAS of the respective scheduler. Figures 4 and 5 depict the execution time of the various approaches, and Figures 6 and 7 present the comparison of design quality with 1% FPTAS solution.

Evaluation of LP-EDF FPTAS The run time and memory usage of the optimal technique increases exponentially, and we were unable to obtain the results in a reasonable amount of time for task sets greater than 70 nodes. Both SGA and EGA are quite efficient and can generate results for large task sets (100 nodes) in less than a

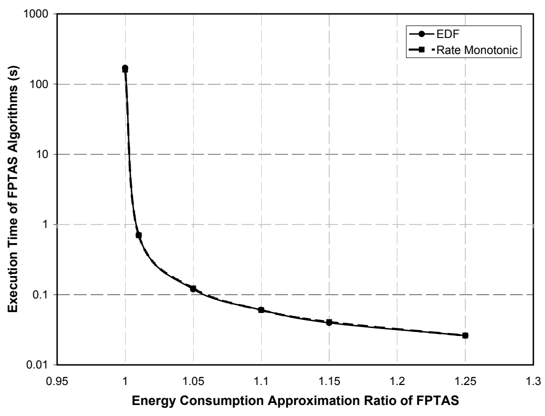

Figure 8: Execution time versus design quality

second[2]. The execution time of LP-EDF FPTAS is comparable to SGA and EGA for a quality bound greater than 5%. The LP-EDF FPTAS with a quality bound of 1% takes just over a second for 90 and 100 node task sets. The design quality of SGA and EGA was on an average inferior to LP-EDF FPTAS with a quality bound of 25%. For quality bounds of less than or equal to 15% the LP-EDF FPTAS was much superior to both SGA and EGA techniques.

Evaluation of LP-RM FPTAS The run time of the optimal technique for LP-RM rises much fast than LP-EDF. Both SGA and EGA are very efficient, and are comparable to run time of LP-RM FPTAS with a quality bound of 25%. LP-RM FPTAS with a quality bound of 1% requires just over a second for large 80 to 100 node task sets. The run times of LP-RM FPTAS for all other quality bounds was under a second for large task sets. The design quality of SGA and EGA was comparable to LP-RM FPTAS with a quality bound of 25%. The LP-RM FPTAS was much superior to both SGA and EGA techniques for quality bounds of less than or equal to 15%.

Summary Although SGA and EGA are efficient techniques, the average quality of their solutions is consistently poorer than LP-EDF and LP-RM FPTAS for quality bounds less than or equal to 15%.

5.3 Execution time versus quality bounds

Figure 8 plots the average execution time of the LP-EDF and LP-RM FPTAS for synthetic task sets versus the approximation ratio or quality bound[3]. We can observe that at a quality bound of 10% the execution time of the approaches is less than 0.001 times the run time of the optimal. Thus, at 10 % quality bound the two approaches offer an excellent trade-off between design quality and solution time.

6. CONCLUSION

We addressed the minimum power consumption assignment of voltage/frequency states for a sequence of tasks to be executed on an embedded processor by EDF and RM schedulers. We showed that the problem is NP-hard and presented FPTAS as solutions. Experimental results with both multimedia applications and synthetic benchmarks demonstrate that our approaches with a quality bound of 1% are able to get very close to the optimal, and produce high quality solution even when an approximation bound of 25% is considered.

[2]We do not plot the execution times for SGA and EGA for less than 60 nodes as they are close to zero.

[3]The plots of the two approaches are overlapping. Consequently, the plots appear as only a single curve.

7. REFERENCES

[1] N. K. Jha. Low power system scheduling and synthesis. In *Proceedings of ICCAD*, 2001.

[2] L. Benini and G. De Micheli. A survey of design techniques for system-level dynamic power management. *IEEE Transactions on VLSI Systems*, 2000.

[3] C.L. Liu and J.W. Layland. Scheduling algorithms for multiprogramming in hard-real-time environment. *Journal of ACM*, 20(1), January 1973.

[4] P. Mejia-Alvarez, E. Levner, and D. Mosse. Adaptive scheduling server for power aware real-time tasks. *ACM TECS*, 3(2):284–306, May 2004.

[5] F. Yao, A. Demers, and S. Shenker. A scheduling model for reduced cpu energy. In *IEEE Annual Foundations of Comp. Sci.*, 1995.

[6] T. Ishihara and H. Yasuura. Voltage scheduling problem for dynamic variable voltage processors. In *Proceedings of ISLPED*, 1998.

[7] Y. Shin, K. Choi, and T. Sakura. Power opimization of real-time embedded systems on variable speed processors. In *Proceedings of ICCAD*, 2000.

[8] P. Pillai and K. G. Shin. Real-time dynamic voltage scaling for low-power embedded operating systems. In *Proceedings of ACM Symposium on Operating Systems Principles*, 2001.

[9] H. Aydin, R. Melhem, D. Mosse, and P. M. Alvarez. Dynamic and aggressive scheduling techniques for power aware real-time systems. In *Proceedings of the IEEE Real-Time Systems Symposium*, 2001.

[10] L. Yan, J. Luo, and N. K. Jha. Combined dynamic voltage scaling and adaptive body biasing for heterogeneous distributed real-time embedded systems. In *Proceedings of ICCAD*, 2003.

[11] S. Irani, S. Shukla, and R. Gupta. Algorithms for power savings. In *Proceedings of the 14th Symposium on Discrete Algorithms*, 2003.

[12] R. Jejurikar, C. Pereira, and R. Guptar. Leakage aware dynamic voltage scaling for real-time embedded systems. In *Proceedings of DAC*, 2004.

[13] Y. Shin, K. Choi, and T. Sakura. Dynamic voltage and frequency scaling under a precise energy model considering variable and fixed components of the system power dissipation. In *Proceedings of ICCAD*, 2004.

[14] B. Mochocki, X. S. Hu, and G. Quan. A unified approach to variable scheduling for nonideal dvs processors. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23(9), September 2004.

[15] F. Xie, M. Martonosi, and S. Malik. Bounds on power savings using runtime dynamic voltage scaling: An exact algorithm and a linear-time heuristic approximation. In *Proceedings of ISLPED*, 2005.

[16] J. Chen, T. Kuo, and C. Shih. $(1+\epsilon)$ approximation clock rate assignment for periodic real-time tasks on a voltag-scaling processor. In *Proceedings of EMSOFT*, 2005.

[17] X. Zhong and C. Xu. System-wide energy minimization for real-time tasks: Lower bound and approximation. In *Proceedings of ICCAD*, 2006.

[18] P. Yang and F. Catthoor. Pareto-optimization-based run-time task scheduling for embedded systems. In *Proceedings of CODES+ISSS*, 2003.

[19] W. Kim, J. Kim, and S. L. Min. A dynamic voltage scaling algorithm for dynamic-priority hard real-time systems using slack time analysis. In *Proceedings of DATE*, 2002.

[20] C. M. Krishna and Y. H. Lee. Voltage-clock-scaling adaptive scheduling techniques for low power in hard real-time systems. *IEEE Transactions on Computers*, 52(12), December 2003.

[21] Y. Zhu and F. Mueller. Feedback edf scheduling exploiting dynamic voltage scaling. In *Proceedings of RTAS*, 2004.

[22] J. Zhuo and C. Chakrabarti. System-level energy-efficient dynamic task scheduling. In *Proceedings of DAC*, 2005.

[23] B. Mochocki, X. S. Hu, and G. Quan. Practical on-line dvs scheduling for fixed-priority real-time systems. In *Proceedings of RTAS*, 2005.

[24] V. Swaminathan and K. Chakrabarty. Network flow techniques for dynamic voltage scaling in hard real-time systems. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 2004.

[25] Vijay V. Vazirani. *Approximation Algorithms*. Springer-Verlag, 2001.

[26] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer-Verlag, 2004.

[27] A. K. Chandra, D. S. Hirschberg, and C. K. Wong. Approximation algorithms for some generalized knapsack problems. *Theoretical Computer Science*, (3):293–304, 1976.

[28] E. L. Lawler. Fast approximation algorithms for knapsack problems. *Mathematics of Operations Research*, (4):339–356, 1979.

[29] K. Niyogi and D. Marculescu. Speed and voltage selection for gals systems based on voltage/frequency islands. In *Proceedings of ASPDAC*, 2005.

[30] A. Sinha and A. P. Chandrakasan. Jouletrack-a web based tool for software energy profiling. In *Proceedings of DAC*, 2001.