Checkpoint Selection for DEPS Framework Based on Quantitative Evaluation of DEPS Profile

Hirotaka Kawashima, Gang Zeng, Hideki Takase, Masato Edahiro and Hiroaki Takada
Graduate School of Information Science, Nagoya University
Nagoya, Japan

Abstract— A dynamic energy performance scaling (DEPS) framework had been proposed as a generalization of the dynamic voltage frequency scaling (DVFS). In this paper, we propose a scheme of checkpoint selection for DEPS framework. The checkpoint is a inserted function call in a program for switching the hardware configurations. Our scheme of checkpoint selection judges energy efficiency of a checkpoint set using intra-task analysis informations. This information is called the DEPS profile. It consists of sets of hardware configurations, execution time and energy consumption of a task. Our scheme evaluates DEPS profiles related with different checkpoint sets, and determines which checkpoint set is the most energy efficient. To achieve this goal, we also propose a quantitative evaluation method of the DEPS profile. This method enables us to judge which DEPS profile is the most energy efficient. From experimental results, we confirm the reasonability of our quantitative evaluation, and that our scheme can select the optimal checkpoint set in realistic time.

I.. Introduction

Power and energy consumption has become one of the major concerns in today's embedded system design. Reducing energy consumption can extend battery lifetime of portable systems, decrease chip cooling costs, as well as increase system reliability. Various techniques have been proposed to optimize the energy consumption of embedded systems so far. Dynamic voltage and frequency scaling (DVFS) is one of major techniques to optimize the energy consumption. This technique controls supply voltage and operating frequency of a system. However, as the feature size of VLSI consistently shrinks, the voltage of a processor has become so low that the room for DVFS has been limited.

Dynamic energy performance scaling (DEPS) have been proposed[1], and integrated into a framework for energy optimization of embedded real-time applications[2]. DEPS is a generalized concept of DVFS. DEPS aims to control not only operation frequency and supply voltage, but also the other reconfigurable hardware resources such as the number of cache ways. The rationale of DEPS is mainly based upon a fact that energy and performance trade-off exists in many hardware resources. Specifically, the higher the processor performance is (i.e., means the shorter execution time), the more energy the processor will consume to execute a program. The basic idea of DEPS framework is to use the processor configuration of minimum performance to save the maximum energy while meeting all real-time constraints.

Similarly to many DVFS systems, DEPS switches hardware configurations in the function called *checkpoints*[3]. However, checkpoint process consumes both energy and execution time for calculation and transition of hardware configuration. The amount of overheads increase as the number of checkpoints increases. Therefore, the number of inserted checkpoints in a program should be limited, and only more energy efficient checkpoints should be used for energy optimization. Generally, timing of switching hardware configurations has large impact on energy consumption of entire system. Therefore, it is very important for energy optimization to determine where to insert checkpoints. Many techniques for extracting the location of checkpoints are proposed for DVFS systems [4, 5, 6, 7]. Many of the DVFS methods formulate the relation between execution time and energy consumption. Also they extract checkpoints according to different DVFS strategies. Namely their extraction methods depend on their DVFS policy by which frequency and voltage are to be assigned. In this paper, we propose a scheme of checkpoint selection for DEPS framework due to the following two reasons:

- Because DEPS abstracts the hardware resources, it is difficult to formulate the relation between execution time and energy consumption at a given hardware configuration.
- Because the existing checkpoint extraction methods is strongly related with their DVFS strategy, they are not suitable for DEPS.

The proposed scheme aims to select an energy efficient checkpoint set from given checkpoint candidates. We make multiple checkpoint sets from checkpoint candidates, and compare their energy efficiency. We use DEPS profiles as indexes of energy efficiency of the task with each checkpoint set. The DEPS profile is one of outputs of the intra-task optimization phase of DEPS framework. Because more energy efficient checkpoint set indicates more energy efficient DEPS profile, the proposed scheme selects the most energy efficient checkpoint set by evaluating DEPS profiles. For this purpose, we propose a quantitative evaluation method for DEPS profiles. The proposed method compares the energy efficiency of multiple DEPS profiles, and determine which profile is more energy efficient.

The contributions of this paper are as follows:

Checkpoint Selection based on Profile Evaluation (CSPE) A scheme of checkpoint selection for DEPS framework is proposed. The proposed scheme is the first checkpoint selection method for DEPS framework. Because DEPS is generalization of DVFS, this scheme is also applicable to DVFS systems. A greedy algorithm based on the CSPE can select checkpoint set with higher energy efficiency over existing checkpoint selective method in realistic time.

Quantitative DEPS Profile Evaluation (QDPE) A quantitative evaluation method of DEPS profiles is also proposed.

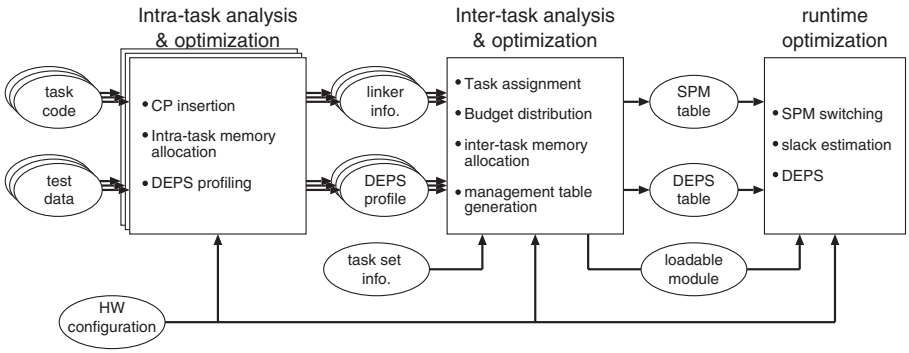


Fig. 1. The overview of integrated energy optimization framework [2]

This evaluation method enables us to evaluate the energy efficiency of a task, and determine which profile is more energy efficient.

This paper is organized as follows: the overview of DEPS framework is introduced in Section II.. We propose CSPE scheme in Section III.. Section IV. proposes QDPE, which is required in CSPE scheme. We show a checkpoint selection algorithm based on CSPE scheme in Section V.. Section VI. shows experimental results, and Section VII. concludes this paper.

II.. Dynamic Energy/Performance Scaling Framework

The objective of DEPS is to minimize the energy consumption in the average case for an embedded real-time application. The DEPS framework consists of several software analysis tools and an energy-aware real time operating system (RTOS). Fig. 1 denotes the workflow of the integrated framework [2] which includes both DEPS and a memory management functions. As can be seen, it consists of three phases. In this section, we only give a brief introduction to the DEPS-related contents, and more details can be found in [2].

The first is an intra-task optimization phase. This phase mainly performs checkpoint insertion and generation of DEPS profile for each task. CSPE scheme can applied to DEPS framework here. The DEPS profile is generated after insertion of checkpoints, and consists of the following information:

- **configuration set** A combination of hardware configurations of all checkpoints.
- **WCET** Worst-Case Execution Time of a task when the configuration set is selected.
- **AEC** Average Energy Consumption of multiple input data of a task when the configuration set is selected.

We show an example of the DEPS profile in Table I. WCET and AEC in the DEPS profile must be Pareto optimal regarding energy efficiency of a task. The DEPS profile plays an important role in this paper.

The second is an inter-task optimization phase. This phase performs multi-task energy optimization by considering all task-related information such as period, deadline, etc. Specifically, execution time budgets are distributed to each task in such a way that the total system energy is minimized and all deadlines are met. To this end, an integer linear problem is constructed with

TABLE I
An example of the DEPS profile

configuration set				WCET	AEC
CP0	CP1	CP2	CP3		
cfg1	cfg1	cfg2	cfg1	37.000	35.125
cfg2	cfg1	cfg2	cfg1	38.000	33.500
cfg2	cfg1	cfg2	cfg2	40.000	33.250
cfg1	cfg2	cfg2	cfg1	46.000	33.000
cfg2	cfg2	cfg2	cfg1	48.000	31.375
cfg2	cfg2	cfg2	cfg2	50.000	31.125

the task set information and the DEPS profiles of each task[1]. As a result, the assigned execution time budget and DEPS profile of each task are used to determine the default configuration set of a task, and DEPS management tables are generated for each checkpoint.

The final is a runtime optimization phase. A DEPS enabled RTOS was developed as an extended version of TOPPERS/ASP kernel [8]. Basically, the RTOS switches the assigned default configuration at each checkpoint during runtime according to the DEPS management table. Additionally, when the RTOS detects dynamic slack, it switches the default configuration to a next one in the table with less energy consumption.

As can be seen, a significant advantage of the DEPS framework is that it can not only apply static and runtime optimization but also apply intra-task and inter-task optimization to achieve the maximum energy savings. A case study using real applications and the prototype DEPS-ready processor [9] has shown that significant energy savings can be achieved by the framework [2].

III.. Checkpoint Selection Scheme for DEPS Framework

In this section, we propose CSPE scheme for DEPS framework. CSPE scheme assumes that the candidates of checkpoints and desirable number of checkpoints for insertion have been given, and aims to select an energy efficient checkpoint set from the candidates.

We focus on DEPS profile, and regard it as an indicator of energy efficiency. As described in Section II., the DEPS profile is one of outputs of the intra-task optimization phase, and contains energy characteristics of tasks. Therefore we expect to know the energy efficiency of a task by analyzing its DEPS profile. Furthermore we expect to evaluate intra-task optimization methods by analyzing each DEPS profile generated from each optimization methods. For checkpoint selection, we prepare multiple checkpoint sets from the checkpoint candidates. DEPS

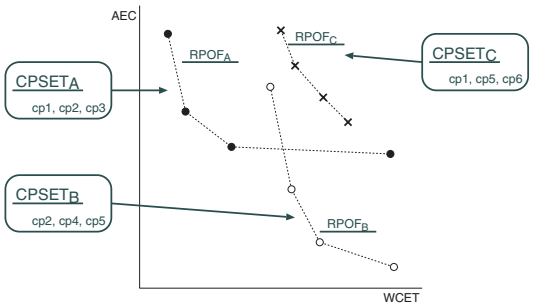


Fig. 2. DEPS profiles of a same task provided by different checkpoint sets

profiles are generated using each checkpoint set. DEPS profiles are analyzed and compared, then the most energy efficient DEPS profile is determined. Meanwhile, the corresponding checkpoint set can be selected as the most energy efficient one.

We describe CSPE scheme for DEPS framework using an example shown in Fig. 2. Fig. 2 plots three DEPS profiles. Each plotted point in the figure corresponds to a WCET, AEC, and configuration set. The target task already has six checkpoint candidates $\{cp1, cp2, cp3, cp4, cp5, cp6\}$. Here we suppose to select three checkpoints from six candidates. Briefly, the CSPE mainly includes the following four steps.

1. Prepare checkpoint sets with specified number of checkpoints from checkpoint candidates.

   In Fig. 2, we prepare three checkpoint sets $CPSET_A = \{cp1, cp2, cp3\}$, $CPSET_B = \{cp2, cp4, cp5\}$ and $CPSET_C = \{cp1, cp5, cp6\}$. Note that in this step, the checkpoint sets can be selected by any heuristic or exhaustive methods. We show selection method based on greedy algorithm in Section V.. Additionally we use an algorithm based on exhaustive search in Section VI..

2. Generate DEPS profiles for each checkpoint set.

   Three DEPS profiles $PROF_A$, $PROF_B$ and $PROF_C$ in Fig. 2 are generated corresponding to $CPSET_A$, $CPSET_B$ and $CPSET_C$, respectively.

3. Compare the energy efficiency of the DEPS profiles, and specify the most energy efficient DEPS profile.

   $PROF_C$ is obviously inferior to $PROF_A$ and $PROF_B$. On the other hand, for $PROF_A$ and $PROF_B$, it is difficult to judge which profile is superior from visuals of them. For evaluating and comparing DEPS profiles, We propose QDPE method in Section IV.. This method evaluates the energy efficiency based on the average value of the smallest realizable energy consumption under the assumption of uniform distribution of time budgets. QDPE enables us to determine which profile is more energy efficient.

4. Determine the most energy efficient checkpoint set.

   We believe that the most energy efficient checkpoint set can provide the most energy efficient DEPS profile. Therefore, we can maintain the checkpoint set $CPSET_C$ is inferior to $CPSET_A$ and $CPSET_B$. We expect to determine the energy efficient checkpoint set between $CPSET_A$ and $CPSET_B$ by QDPE method for DEPS profile.

As mentioned above, some details should be discussed further to apply the CSPE. Section IV. will presents a quantitative evaluation method for DEPS profile, and Section V. will give a greedy algorithm based on the CSPE.

IV.. Quantitative DEPS Profile Evaluation Method

A.. An Evaluation Method for Single DEPS Profile

In this section, we describe how to score single DEPS profile.

Firstly, we define the smallest realizable energy consumption. The DEPS profile consists of Pareto optimal sets of WCET, AEC and configuration set. However, we can not realize all configuration sets due to deadline constraint of a task. We can realize only configuration sets such that its WCET is smaller than the time budget of a task. Otherwise, the configuration set misses the deadline of a task. Therefore, when time budget is given, we can know the smallest realizable energy consumption. We suppose that DEPS profile composed of $(t_0, e_0), (t_1, e_1), \cdots, (t_n, e_n)$ is given, and $t_0 < t_1 < \cdots < t_n$. When time budget b is given, the smallest realizable energy consumption $srec(b)$ can be calculated as follows:

$$srec(b) = \begin{cases} none & \texttt{if } b < t_0 \\ e_i & \texttt{if } t_i \leq b \texttt{ and } b < t_{i+1} \\ e_n & \texttt{if } t_n \leq b. \end{cases} \quad (1)$$

Fig. 3 shows an example of a plot of DEPS profile and the smallest realizable energy consumption for each time budget. When a time budget t_b is assigned between t_3 and t_4, the smallest realizable energy consumption is $srec(t_b) = e_3$.

From the point of optimizing the energy consumption, DEPS profile with the smaller realizable energy consumption is more superior to the others. Unfortunately, the time budget is given in the inter-task optimization phase. Therefore, the smallest realizable energy consumption for a specific time budgets is not available in the intra-task phase. We need to determine the superiority of the DEPS profiles without fix the time budget to specific value. To determine the superiority without fixing the time budget, we use the area below the smallest realizable energy consumption for each time budget. We show the area which corresponds to the evaluation value in Fig. 3 as a shaded area. The DEPS profile is a Pareto optimal sets of WCET and AEC. More energy efficient DEPS profiles is plotted nearer to the origin. The DEPS profile nearer to the origin is expected to indicate smaller area. Therefore, we can judge the DEPS profile with the smaller area is superior to the other DEPS profiles. We can find that the shaded area in Fig. 3 is the sum of the realizable smallest energy consumption. The score is proportional to the average of $srec(t_b)$ under the assumption that time budgets t_b are distributed uniformly. Therefore, it is reasonable to judge that the DEPS profile which has smaller area is more energy efficient.

Next, we define a section for scoring. The lower bound of the available time budgets is equal to the smallest WCET of the DEPS profile. Even if time budget that is smaller than the smallest WCET in the DEPS profile is given, there is no configuration sets that can meet the deadline constraint. Because time budget under the smallest WCET is invalid, we do not consider these section in scoring DEPS profile. Then we define the smallest WCET in the DEPS profile as the lower bound of scoring section. If a time budget that is larger than the largest WCET in DEPS profile is given, any configuration sets can satisfy the deadline constraints. Therefore, there is no upper bound of the available time budgets. However, DEPS profile must be scored in a finite

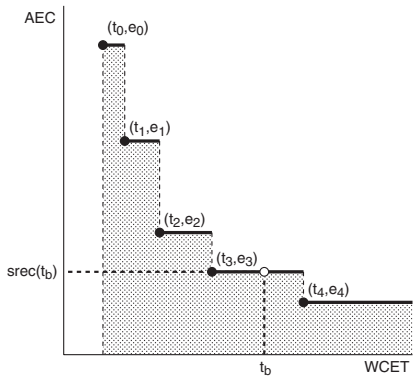


Fig. 3. The smallest realizable energy consumption

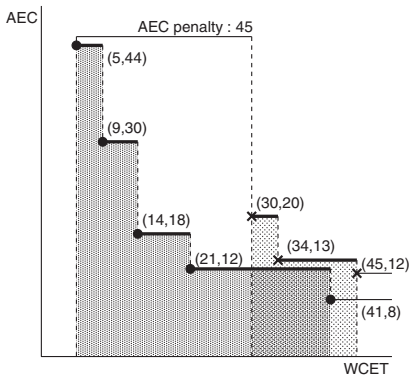


Fig. 4. An example of score inversion

section, we need to define the upper bound of the scoring section. In this paper, we use the largest WCET in the DEPS profile as the upper bound of scoring section.

Based on these discussion, we define the score of a DEPS profile. We assume a DEPS profile consists of $(t_0, e_0), (t_1, e_1), \cdots, (t_n, e_n)$. The scoring section for this DEPS profile is $[t_0 : t_n]$. We define the score of the DEPS profile as follows:

$$score = \frac{\sum_{i=0}^{n-1} e_i(t_{i+1} - t_i)}{t_n - t_0}. \quad (2)$$

B.. Comparison of Multiple DEPS Profiles

In this subsection, we describe the comparison of multiple DEPS profiles.

In scoring single DEPS profile, we use the largest WCET and the smallest WCET as the upper bound and the lower bound, respectively. When comparing multiple DEPS profiles, we also use this policy. However, scoring sections for multiple DEPS profiles may be different. Even if DEPS profiles are scored for each different scoring section, we can not compare them and can not determine which profile is superior. We show an example of score inversion caused from difference of scoring sections in Fig. 4. Fig. 4 shows two DEPS profiles indicated by bullets and cross marks. When we judge visually, the DEPS profile with bullets is obviously superior to that with cross marks. On the other hand, when we score the DEPS profiles using Eq. 2, the score of the DEPS profile with bullets and cross marks is 19.22 and 14.87, respectively. The evaluation using the score calculated by Eq. 2 is different from the visual evaluation. This example indicates that comparison based on Eq. 2 is incorrect.

To compare multiple DEPS profiles correctly, we need to use the unified scoring sections for the comparison targets. We define the unified scoring section as a section between the smallest WCET and the largest WCET of all DEPS profiles to be compared. When the unified scoring section is employed, we need to score a DEPS profile for the section under its smallest WCET and the section over its largest WCET. Because time budget which is larger than the largest WCET in the DEPS profile is valid, we can score using the largest WCET. When the smallest WCET in a DEPS profile is larger than the lower bound of scoring section, we have to score the DEPS profile for invalid section of time budget. We define an AEC penalty value for the invalid section. We calculate the score of DEPS profile using AEC penalty value for the invalid section. If the time budget of a task is assigned to the invalid section, the task can not satisfy the deadline constraint. Therefore, AEC penalty is desired to be large value. On the other hand, too large AEC penalty results pessimistic evaluation. It is not preferable for energy optimization. Therefore, a realistic and large value is required as the AEC penalty. In this paper, we use the largest AEC of all available configuration sets as the AEC penalty value. Intrinsically a task has c^n configuration sets for the number of checkpoints n and the number of configurations c. Because the DEPS profile has Pareto optimal sets of WCET and AEC, a lot of configuration sets are ignored. The AEC penalty value we use in this paper is the largest AEC value among all available configuration sets including these ignored configuration sets. The largest AEC can be calculated deterministically. We suppose the scoring section $[t_s : t_l]$ and the AEC penalty e_p. t_s and t_l indicate the smallest and the largest WCET of target DEPS profiles, therefore $t_s \leq t_0$ and $t_n \leq t_l$, We can calculate the score of a DEPS profile for comparing multiple DEPS profiles as follows:

$$score = \frac{e_p(t_0 - t_s) + \sum_{i=0}^{n-1} (e_i(t_{i+1} - t_i)) + e_n(t_l - t_n)}{t_l - t_s} \quad (3)$$

We evaluate and compare two DEPS profiles shown in Fig. 4. The unified scoring section is $[5 : 45]$. Time budget in section $[5 : 30]$ for DEPS profile with cross marks is invalid. Therefore, we suppose that the AEC penalty 45 is given. The score of DEPS profiles with bullets and cross marks is 18.1 and 33.7, respectively.

We note about cases that it is difficult to decide which DEPS profile is more energy efficient, such as $PROF_A$ and $PROF_B$ in Fig. 2. We assume that checkpoint insertion is done in the intra-task optimization phase. We can not know the time budget which is required to decide the realizable energy consumption in this design phase. Therefore no one can make accurate decision which profile is more energy efficient. While QDPE method also can not make accurate decision, it can make a reasonable decision because it use the average of the smallest realizable energy consumption as a foundation of decision.

V.. A Checkpoint Selection Algorithm Based on CSPE scheme

CSPE scheme enables us to compare and evaluate energy efficiency of checkpoint sets. If we select n checkpoints among m checkpoint candidates, ${}_m\mathrm{C}_n$ checkpoint sets need to be investi-

Algorithm 1 A checkpoint selection algorithm based on greedy approach

Input: $cpcnt$, $cp\_candidates$
Output: $cpset$
1: $cpset = \emptyset$
2: **while** $sizeof(cpset) < cpcnt$ **do**
3: $profs = \emptyset$
4: **for** $j = 0$ to $sizeof(cp\_candidates)$ **do**
5: /* 1. Prepare checkpoint sets from checkpoint candidates. */
6: $target\_cpset$ = $cpset \cup \{cp\_candidates[j]\}$
7: /* 2. Generate DEPS profiles for each checkpoint set. */
8: $target\_profile$ = **deps_profiling**($target\_cpset$)
9: $profs[j]$ = $target\_profile$
10: **end for**
11: /* 3. Compare the energy efficiency of the DEPS profiles, and specify the most energy efficient DEPS profile. */
12: $best\_score$ = 0
13: **for** $j = 0$ to $sizeof(cp\_candidates)$ **do**
14: $score$ = **score_prof**($profs[j]$)
15: **if** $score < best\_score$ or $best\_score$ = 0 **then**
16: $best\_cp$ = $cp\_candidates[j]$
17: $best\_score$ = $score$
18: **end if**
19: **end for**
20: /* 4. Determine the most energy efficient checkpoint set. */
21: $cpset$ = $cpset \cup \{best\_cp\}$
22: $cp\_candidates$=$cp\_candidates \setminus \{best\_cp\}$
23: **end while**
24: **return** $cpset$

gated. Therefore, finding an optimal checkpoint set belongs to NP-hard problems.

To find energy efficient checkpoint sets in realistic time, we introduce a checkpoint selection heuristic based on CSPE scheme in Algorithm 1. We show corresponding operation of CSPE scheme in Section III. as comments. This algorithm is based on the greedy algorithm. The algorithm assumes that $cp\_candidates$ and $cpcnt$ are given as inputs. $cp\_candidates$ is a list of checkpoint candidates. $cpcnt$ is the number of checkpoints to be selected. The algorithm select a set of energy efficient checkpoints $cpset$ from $cp\_candidates$. The algorithm employs greedy approach, namely select energy efficient checkpoints one by one greedily. We create a target checkpoint set $target\_cpset$ as the sum of $cpset$ and a checkpoint candidate cp(line 6). The DEPS profile for $target\_cpset$ is generated by **deps_profiling**(line 8). Details of **deps_profiling** is described in [10]. $prof$ is appended to the set of DEPS profiles $profs$ (line 9). line 6,8,9 are repeated for all checkpoint candidates. Namely, $profs$ consists of DEPS profiles generated from checkpoint sets which a checkpoint candidate is appended to fixed checkpoint list. Next **score_profile** evaluates and compares DEPS profiles in $profs$, specify the most energy efficient checkpoint $best\_cp$ (line 13 to 19). Because we can determine that a checkpoint set related with the most energy efficient DEPS profile is the most energy efficient, we add $best\_cp$ to $cpset$(line 21), and exclude $best\_cp$ from $cp\_candidates$. These operations are repeated until the number of the fixed checkpoint reaches $cpcnt$.

VI. Evaluation

A. Experimental Setup

We develop a brief version of the video-conference system as the target application [2]. The target application consists of video encoding task and decoding task. We used the Xvid MPEG-4 video codec [11] and the FFmpeg library [12]. We assume the prototype reconfigurable processor [9] which provides two-way operating frequency and four-way variable cache ways, i.e., eight hardware configurations. We extract twenty checkpoint candidates by checkpoint extraction technique for DVFS system [13]. This method puts checkpoints on locations where remaining worst-case execution time of a task changes greatly. Namely, checkpoint candidates are located after branches in application programs. We select eight checkpoints within twenty checkpoint candidates. We compare following four checkpoint selection methods.

random This method selects eight checkpoints randomly within twenty checkpoint candidates. Because there are no existing checkpoint selection method for DEPS framework, we use this method as a target method of comparison in this paper.

rankedDVFS Twenty checkpoint candidates are ranked by the checkpoint extraction and ranking method for DVFS systems [13], and top eight checkpoints are selected. This method ranks checkpoint under DVFS assumptions, which considers switching only operating frequencies. It is unfair to compare with algorithms based on CSPE scheme, which considers switching abstracted hardware resources. However, we have no existing checkpoint selection method for DEPS systems, therefore we employ this method as an existing method to select desired number of more energy efficient checkpoints.

exhaustive All combinations of eight checkpoints within twenty checkpoint candidates are investigated by the exhaustive search. We investigate all combinations , and pick up the most energy efficient checkpoint set. Namely the output checkpoint set is optimal one. Because this method employs QDPE, this method is based on CSPE scheme,

greedy A checkpoint selection algorithm shown in Algorithm 1. This method is also based on CSPE scheme.

We implement checkpoint selection algorithms with Perl v5.10.1, and executed on Xeon X5680 3.33GHz, 65GB memory system.

B. Experimental Results

We show processing time of checkpoint selection and scores of resulting DEPS profiles in Table II. Here we show three checkpoint sets selected by **random** for video decoding and encoding task, respectively. We also show the DEPS profiles related with the checkpoint sets selected by **random**, **rankedDVFS**, **exhaustive** and **greedy** in Fig. 5. Note that DEPS profiles provided by **exhaustive** and **greedy** completely match, therefore **exhaustive** does not appear in Fig. 5.

As a contribution of this paper, we can judge which DEPS profile is more efficient by QDPE method, while we could judge only by visual effects without our contribution. From visual effects in Fig. 5, we can judge the DEPS profiles provided by **exhaustive** and **greedy** are more energy efficient than **random**s and **rankedDVFS**. We can make the same judgment from the scores of DEPS profiles in Table II, because the scores of **exhaustive** and **greedy** are the smallest. This result shows that CSPE scheme can select energy efficient checkpoint sets for DEPS framework. We have cases that it is difficult to decide which DEPS profile is more energy efficient, such as **random1** and **random2** in Fig. 5(a), and **random5** and **random6** in Fig. 5(b). QDPE method can decide which profile is more energy efficient based on the average of the smallest realizable energy consumption. In Fig. 5(a)

TABLE II
PROCESSING TIME OF CHECKPOINT SELECTION AND SCORE OF PROVIDED DEPS PROFILES

task	algorithm	time	score	rate
decode	exhaustive	2229 min	1073537.871	-
	greedy	80.78 sec	1073537.871	0%
	random1	-	1142509.979	+6.424%
	random2	-	1143830.856	+6.547%
	random3	-	1086187.085	+1.178%
	rankedDVFS	-	1088066.400	+1.353%
encode	exhaustive	20273 min	8311303.265	-
	greedy	306.55 sec	8311303.265	0%
	random4	-	8383340.859	+0.867%
	random5	-	8382074.043	+0.852%
	random6	-	8393933.178	+0.994%
	rankedDVFS	-	8433738.787	+1.473%

and Fig. 5(b), we can decide **random1** and **random5** is more energy efficient, respectively.

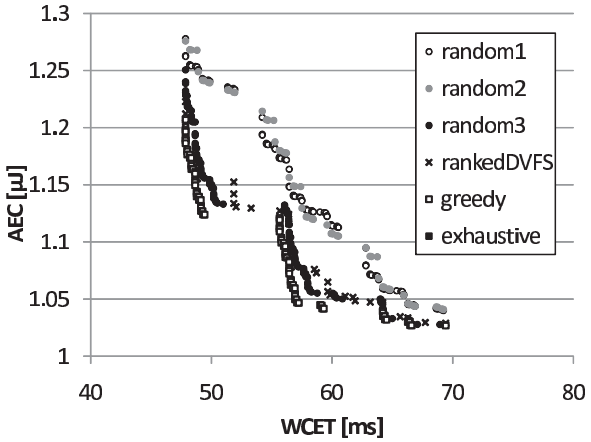
We also show the usefulness of **greedy** by comparing with **exhaustive**. While the checkpoint set selected by **exhaustive** is optimal, **exhaustive** takes so long time that we can not use it practically. On the other hand, **greedy** can select the checkpoint set in realistic time. It is highly important result that checkpoint sets selected by **greedy** are the same as **exhaustive** for both decoding and encoding task. Namely the checkpoint set obtained from **greedy** are also optimal ones. and DEPS profiles provided by **greedy** is the same as that from **exhaustive** as shown in Fig. 5. From this result, we consider that **greedy** is suitable for DEPS framework, and we can confirm the usefulness of **greedy**.

VII.. CONCLUSION

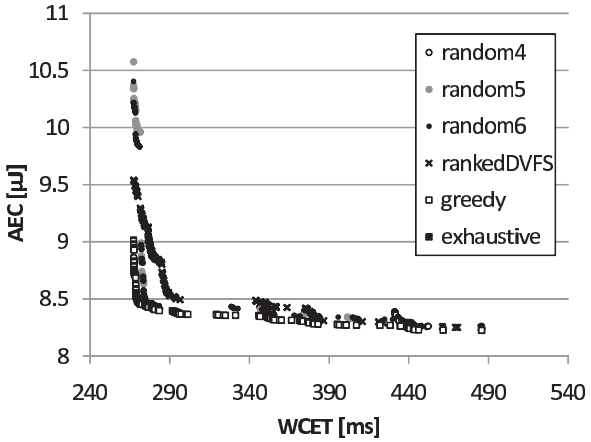
In this paper, we propose CSPE scheme, which provides checkpoint selection for DEPS framework. We use DEPS profiles as indexes of energy efficiency of checkpoint sets. We compare multiple DEPS profiles, and select checkpoint set related with the most energy efficient DEPS profile. To achieve CSPE scheme, we need to compare the DEPS profiles. Then we also propose QDPE method, which evaluate energy efficiency of DEPS profiles. QDPE method use the average of AEC under assumption that time budgets are distributed uniformly as an evaluation value. We show an algorithm based on CSPE scheme. We confirm that the algorithm can select more energy efficient checkpoint set in realistic processing time than existing methods. CSPE scheme is expected to be applicable for DVFS systems, because DEPS is an expanded concept of DVFS.

ACKNOWLEDGMENT

This work is supported by Core Research for Evolutional Science and Technology (CREST) of Japan Science and Technology Agency.



(a) Video decoding task



(b) Video encoding task

Fig. 5. DEPS profiles provided from various checkpoint sets

REFERENCES

[1] G. Zeng, H. Tomiyama, and H. Takada, "A generalized framework for energy savings in hard real-time embedded systems," *IPSJ Trans. on Systems LSI Design Methodology*, vol. 2, pp. 167–179, Aug 2009.

[2] H. Takase, G. Zeng, L. Gauthier, H. Kawashima, N. Atsumi, T. Tatematsu, Y. Kobayashi, S. Kohara, T. Koshiro, T. Ishihara, H. Tomiyama, and H. Takada, "An integrated optimization framework for reducing the energy consumption of embedded real-time applications," in *ISLPED*, Aug 2011.

[3] A. Azevedo, I. Issenin, R. Cornea, R. Gupta, N. Dutt, A. Veidenbaum, and A. Nicolau, "Profile-based dynamic voltage scheduling using program checkpoints," in *In Proceedings of DATE*, 2002, pp. 168–175.

[4] D. Shin and J. Kim, "A profile-based energy-efficient intra-task voltage scheduling algorithm for real-time applications," in *ISLPED*, E. Macii, V. De, and M. J. Irwin, Eds. ACM, 2001, pp. 271–274.

[5] D. Shin, J. Kim, and S. Lee, "Intra-task voltage scheduling for low-energy, hard real-time applications," *IEEE Design & Test of Computers*, vol. 18, no. 2, pp. 20–30, 2001.

[6] J. Seo, T. Kim, and K.-S. Chung, "Profile-based optimal intra-task voltage scheduling for hard real-time applications," in *Proceedings of the 41st Annual conference on Design Automation (DAC-04)*. New York: ACM Press, Jun. 7–11 2004, pp. 87–92.

[7] G. S. A. Kumar and G. Manimaran, "An intra-task DVS algorithm exploiting program path locality for real-time embedded systems," in *HiPC*, ser. Lecture Notes in Computer Science, D. A. Bader, M. Parashar, S. Varadarajan, and V. K. Prasanna, Eds., vol. 3769. Springer, 2005, pp. 225–234.

[8] "Toppers project," http://www.toppers.jp/en/.

[9] T. Ishihara, "A multi-performance processor for reducing the energy consumption of real-time embedded systems," *IEICE Transactions*, vol. 93-A, no. 12, pp. 2533–2541, 2010.

[10] H. Kawashima, G. Zeng, N. Atsumi, T. Tatematsu, H. Takase, and H. Takada, "Intra-task analysis of worst case execution time and average energy consumption on deps framework," in *IEICE technical report VLD-432*, 2011, pp. 19–24, (in Japanese).

[11] "Xvid, mpeg-4 compliant video codec," http://www.xvid.org.

[12] "Ffmpeg multimedia system," http://www.ffmpeg.org.

[13] T. Tatematsu, H. Takase, G. Zeng, H. Tomiyama, and H. Takada, "Checkpoint extraction using execution traces for intra-task dvfs in embedded systems," in *The 6th International Symposium on DELTA 2011*, Queenstown, New Zealand, Jan 2011, pp. 19–24.