Vol.2011-SLDM-149 No.4
Vol.2011-EMB-20 No.4
2011/3/18

DEPSプロファイルの評価法とそれを利用したチェックポイント選定

川島裕崇[†1] 曾剛[†2] 渥美紀寿[†1]
立松知紘[†1] 高瀬英希[†1] 高田広章[†1]

本論文ではDEPSフレームワークにおけるタスクの最悪実行時間-平均消費エネルギーをまとめたDEPSプロファイルの定量的評価法を提案する．本論文で提案するDEPSプロファイルの評価手法は，最悪実行時間と平均消費エネルギーをプロットしたときの面積を評価値とする．DEPSプロファイルを評価することでタスク内解析・最適化手法の有効性を評価でき，改善につながる．提案する評価手法を利用したタスク内解析・最適化の改善例として，多数のチェックポイント候補からエネルギー削減効率の高いものを選定する手法と，チェックポイントの通過順を考慮することによるエネルギー削減効率の向上について示す．本論文で提案する評価法を用いることで これらの例で 性能の改善が見られることを示し，本論文の評価法が有効であることを示す．

An Evaluation Method for DEPS Profile and its Application to Checkpoint Selection

Hirotaka Kawashima,[†1] Gang Zeng,[†2]
Noritoshi Atsumi,[†1] Tomohiro Tatematsu,[†1]
Hideki Takase[†1] and Hiroaki Takada [†1]

In this paper, we propose a quantitative evaluation method of a DEPS profile. The DEPS profile is a summarized information of worst-case execution time(WCET) and average energy consumption(AEC) of a task. The proposed method takes a plot area of WCET and AEC as an evaluation value. We can improve the intra-task analysis and optimization using the DEPS profile evaluation. We show two examples of improvement by using the proposed evaluation method. One example is selecting energy efficient checkpoints from many checkpoint candidates. The other example is checkpoint behavior considering the pass order of checkpoint. We show the usefullness of the proposed evaluation method through the two examples.

1. はじめに

近年，組み込みシステムにおいて低消費エネルギー化が強く求められている．消費エネルギーの削減には，携帯機器のバッテリ持続時間の延長，冷却コストやランニングコストの低減など，多くの利点がある．プロセッサの消費エネルギーを削減するための有力な手法として，DVFS(Dynamic Voltage and Frequency Scaling) がある．DVFS は，処理性能が要求される場合には高い電源電圧，動作周波数で処理を行ない，そうでないときには低い電源電圧，動作周波数で処理を行なうことで消費電力，消費エネルギーの低減を図る技術である．我々は DVFS の制御対象を一般化したものとして，DEPS(Dynamic Energy Performance Scaling) を提案している[1)]．DEPS では，性能と消費エネルギーを制御することが可能なプロセッサを用いて，実行時間と消費エネルギーのトレードオフを得る (図 1)．実行時に高い性能が要求される場合には高性能で消費エネルギーの大きいプロセッサ構成で動作し，そうでないときは低性能で消費エネルギーの小さいプロセッサ構成で動作させることを考える．

本稿では，DEPS を用いた低消費エネルギーソフトウェア開発の工程において必要とされる DEPS プロファイルの定量的評価法を提案する．DEPS プロファイルとは，タスク単体の最悪実行時間，平均消費エネルギーについてまとめたものである．タスクに割り当てられるバジェットが確定していれば，DEPS プロファイルを参照することによって実現可能な消費エネルギーが判明するが，本稿で扱う設計フェイズにおいては，バジェットは確定していない．そこで，DEPS プロファイルからバジェットに関する消費エネルギーの平均値を算出し，DEPS プロファイルの評価値とした．提案手法で DEPS プロファイルを評価することにより，DEPS プロファイル同士の優劣をつけることが可能になる．その結果，タスク内解析・最適化が改善された．本稿では，チェックポイント選定と通過順序を考慮したチェックポイントの動作の二つの例を示し，提案手法が有効であることを示す．

本稿の構成は以下のとおりである．2 節で DEPS に基づくソフトウェア開発環境の概要を述べる．3 節で DEPS プロファイルの定量的評価法を示す．4 節で DEPS プロファイル

†1 名古屋大学大学院 情報科学研究科
Department of Information Engineering, Graduate School of Information Science, Nagoya University
†2 名古屋大学大学院 工学研究科
Graduate School of Engineering, Nagoya University

Vol.2011-SLDM-149 No.4
Vol.2011-EMB-20 No.4
2011/3/18

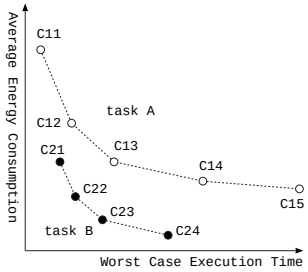


図 1 DEPS における消費エネルギーと実行時間の関係

の評価法を用いて最適化が改善した事例について述べ，5 節で本稿のまとめを行う．

2. DEPS に基づく低消費エネルギーソフトウェア開発環境

2.1 概　　要

本節では，DEPS を利用する低消費エネルギーソフトウェアの開発環境について述べる．本開発環境では，最大応答時間を保証するという制約下で，平均消費エネルギーを最小化するソフトウェアを開発を支援することを目的とする．以下の項目を前提としている．

- ハードリアルタイムシステム
- 性能・消費エネルギーを制御可能なプロセッサ
- ハードウェア構成を反映できる消費エネルギー見積もり環境

本開発環境は，タスク内解析・最適化，タスク間最適化，実行時最適化の 3 段階で消費エネルギーを最適化する．図 2 にソフトウェア開発環境の構成図を示す[2)]．

タスク内解析・最適化においては，まずプロセッサの構成を変更するためのコードを挿入する箇所 (チェックポイント) を決定する．チェックポイントの挿入位置は，命令セットシミュレータを用いて取得したプログラムの実行トレースを解析して決定する[3)]．次に，タスク単体でのメモリ配置の決定を行う．具体的には，タスクの実行トレースからメモリへのアクセス履歴を取り出し，その情報から各メモリ領域を SPM, メインメモリのキャッシュ領域，メインメモリの非キャッシュ領域のいずれに置くかを決定する．さらに，メインメモリのキャッシュ可能領域に置くプログラムを，キャッシュヒット率が上がるように配置する．配置結果は，タスクのメモリ配置情報として，タスク間最適化に渡す．最後に，決定したメモリ配置下で，用意されたテストケースを入力としてタスクの実行トレースを取得し，各チェックポイントで選択するハードウェア構成の組み合わせごとに最悪実行時間と平均消費

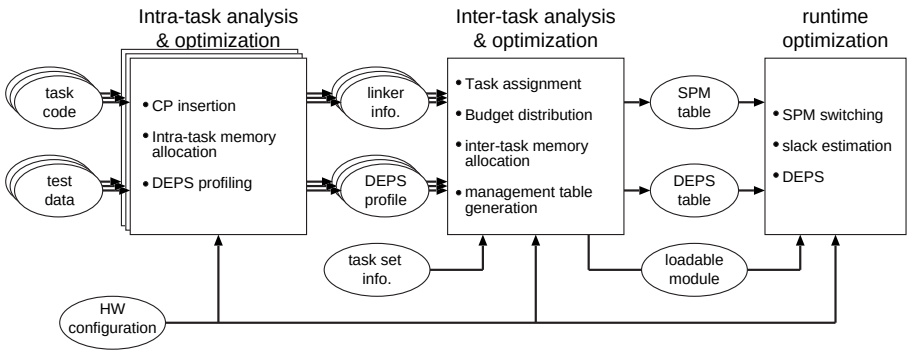


図 2 DEPS に基づくソフトウェア開発環境

エネルギーを算出する．この最悪実行時間と平均消費エネルギーをまとめたものを DEPS プロファイルと呼ぶ[4)]．

タスク間最適化においては，まず各タスクをプロセッサに割り付けると同時に，各タスクに実行時間バジェットを分配する処理を行う．各プロセッサの負荷に余裕が生じた場合，その余裕分をタスクのデッドライン制約が満たされる範囲で，できるだけ低性能で少ないエネルギー消費のハードウェア構成でタスクを動作させる．また，各タスクが使用する SPM の合計容量がプロセッサのもつ SPM の容量よりも大きい場合に，SPM をどのように時分割，空間分割して使用するかを決定し，実行時に参照する SPM 管理テーブルを生成する[5)]．最後に，各タスクの各チェックポイント毎にとりうるハードウェア構成と残り最悪実行時間との関係を示す DEPS 管理テーブルを生成する．

実行時最適化では，各チェックポイントでタスク間最適化で割り当てた実行時間バジェットに対するスラック時間を算出する．デッドライン制約が満たせる最低性能，最低エネルギー消費のハードウェア構成を算出したスラック時間と DEPS 管理テーブルから求め，切り替える．

2.2 DEPS プロファイルの位置づけ

タスク間最適化では，全体として消費エネルギーが最小となるように各タスクに実行時間を割り当てる．その際，各タスクがどの実行時間でどの消費エネルギーで動作するかという情報が必要となる．図 1 ではタスク A とタスク B の最悪実行時間と平均消費エネルギーの関係を示している．この情報を得るには，タスク毎に各チェックポイントで選択するハー

Vol.2011-SLDM-149 No.4
Vol.2011-EMB-20 No.4
2011/3/18

ドウェア構成 (コンフィグレーション) の組み合わせごとに最悪実行時間と平均消費エネルギーを算出し，パレート最適な組み合わせのみを残す．このコンフィグレーションの組み合わせ，最悪実行時間，平均消費エネルギーの情報をまとめたものを DEPS プロファイルと呼ぶ．タスク間最適化では各タスクの DEPS プロファイルを参照し，規定のデッドラインで全タスクの実行が可能な範囲内で消費エネルギーが最小となるようにタスク毎にどのコンフィグレーションの組み合わせを用いるかを決定し，バジェットを割り当てる．

3. DEPS プロファイルの定量的評価法

本節では，DEPS プロファイル同士を比較して優劣を判定するための評価法を提案する．DEPS では，チェックポイントの設定位置が消費エネルギーの削減効率を左右する．チェックポイント処理のオーバヘッドや設計上の制約[4] から設定できるチェックポイントの数には制限があるため，適切な位置を選んでチェックポイントを設定することが重要である．タスク内解析・最適化フェイズにおいてチェックポイントの設定位置を評価するためには，DEPS プロファイルを利用することが考えられる．2.2 節で述べたとおり，DEPS プロファイルはタスクが実現可能な最悪実行時間，平均消費エネルギーのパレート最適解を示すものである．チェックポイントの設定位置としては，DEPS プロファイルがより短い実行時間，少ないエネルギー消費を示すような箇所が望ましい．複数の DEPS プロファイルを比較して優劣をつけることが出来れば，より適切なチェックポイント設定位置の探索が可能になる．DEPS プロファイルに優劣をつけるには，DEPS プロファイルの最悪実行時間，平均消費エネルギーをプロットして人の経験により視覚的に判断するという方法がある．しかし，チェックポイント位置の候補となる場所は多数ある．それらの中からエネルギー効率の優れた位置を選び出すには非常に多くの DEPS プロファイルを比較，評価する必要があり，人が視覚的に評価して低消費エネルギー化の改善につなげるのは困難である．そこで，機械的に DEPS プロファイルの評価を行うための定量的な評価手法が必要となる．

コンフィグレーション $c1, c2, \cdots, cn$ に対して、最悪実行時間が $t_{c1}, t_{c2}, \cdots, tcn$、平均消費エネルギーが $e_{c1}, e_{c2}, \cdots, e_{cn}$ であるような DEPS プロファイルを考える。割り当てられるバジェットが区間 $[t_{c1}, t_{c2}]$ に存在する場合，デッドラインが守られ，かつ消費エネルギーが最小となるのは $c1$ である．そのため，区間 $[t_{c1}, t_{c2}]$ における最小消費エネルギーは e_{c1} となる．このようにバジェットが決まれば，DEPS プロファイルを用いて実現可能な最小の消費エネルギーを算出できる．ただし，バジェットはタスク間最適化が完了するまで決定しないため，チェックポイントの設定箇所を決定するタスク内解析・最適化フェイズでは割り当てられるバジェットの情報を使うことはできない．そこで，割り当てられるバジェットに関する消費エネルギーの区間 $[t_{c1}, t_{cn}]$ における平均値をとって DEPS プロファイルの評価値とする．バジェット割り当てが一様に分布していると仮定すると，区間 t_{c1}, t_{cn} における消費エネルギーの平均値は以下の式で計算できる．

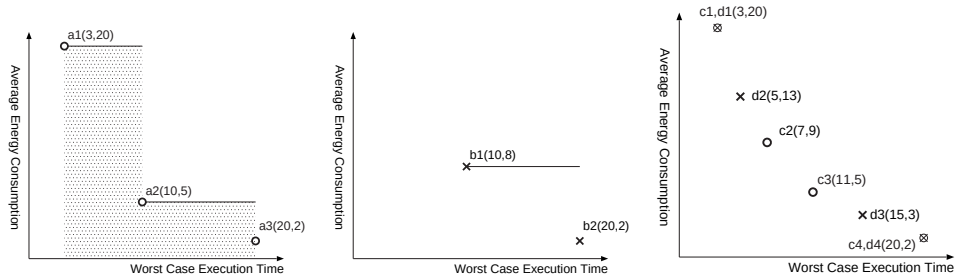
$$E_{ave} = \frac{\sum_{i=1}^{n-1} e_{ci}(t_{ci+1} - t_{ci})}{t_{cn} - t_{c1}} \quad (1)$$

図 3(a) に DEPS プロファイルの例を示す。この DEPS プロファイルでは CP 集合 A として、最悪実行時間と平均消費エネルギーが (3,20)、(10,5)、(20,2) の点が存在する。割り当てられるバジェットとその時の最小の消費エネルギーの関係を実線で示している．この時、区間 $[3, 20]$ の評価値は $\frac{20(10-3)+5(20-10)}{20-3} = 11.2$ となる。

評価値の算出対象となる区間について考える．まず，CP 集合 A 単独の評価値を算出することを考える．割り当てられるバジェットが 3 未満の場合には，タスクはデッドライン制約を満たすことができない．そのため，DEPS プロファイルの評価としては考慮する必要がない．割り当てられるバジェットが 20 以上の場合はとりうる最小の消費エネルギーは 2 となる．このとき，タスクはどのコンフィグレーションを採用してもデッドラインを満たすことができる．評価値を算出するためには有限の区間とする必要があるため，ここでは 20 以上の値も考慮しないこととする．次に，図 3(a) と図 3(b) の評価値を比較する場合を考える．図 3(b) には CP 集合 B として、最悪実行時間、平均消費エネルギーが (10,8)、(20,2) の点が存在する。CP 集合 B は、CP 集合 A よりも明らかに劣っている。ところが、CP 集合 B についても同様に区間 $[10, 20]$ の評価値を計算すると $\frac{8(20-10)}{10} = 8$ となる。評価値は小さいほうが優れていると判断できるため、CP 集合 A よりも優れていると判断されてしまう。これを回避するために，複数の DEPS プロファイルを比較する場合は評価を行う区間を一致させる．評価区間として，比較対象の全ての DEPS プロファイルの最悪実行時間の最小値と最大値を用いる．図 3 の例では，CP 集合 A、CP 集合 B ともに評価区間を $[3, 20]$ として計算を行う．この場合、CP 集合 A の評価値は 11.2、CP 集合 B の評価値は 12.9 で CP 集合 A の方が優れていると正しく判断できる。

提案手法と視覚的な優劣との対比について述べる．DEPS プロファイルはより短い実行時間，少ないエネルギー消費を示すようなものを優れていると判断できる。最悪実行時間、平均消費エネルギーをプロットしたとき、視覚的にはより原点近くにプロットされているものを優れていると判断できる。一方、提案手法では消費エネルギーの平均値を評価値として用いているが、複数の DEPS プロファイルを比較する際には算出区間を一致させる。評

Vol.2011-SLDM-149 No.4
Vol.2011-EMB-20 No.4
2011/3/18



(a) CP 集合 A の DEPS プロファイル (b) CP 集合 B の DEPS プロファイル (c) 優劣がつけづらい場合

図 3 DEPS プロファイルのプロット例

価値の大小は式 1 の分子の値の大小と一致する。式 1 の分子の値は DEPS プロファイルをプロット領域の面積と一致する。図 3(a) では、網掛けで示す領域が式 1 の分子の値と一致する。視覚的な評価と照らし合わせると、より原点近くに点が存在する場合、図 3(a) の網掛けで示す領域も小さくなるため、視覚的に優劣が明らかな場合には提案手法でも正しく優劣を判断できる．提案手法は視覚的に優劣がつけづらい場合にも消費エネルギーの平均値に基づいて優劣をつける事ができる．図 3(c) に CP 集合 C、CP 集合 D の例を示す。$c1$ と $d1$、$c4$ と $d4$ は一致しているとする。このような場合、優劣は決定的ではなく様々な要素について検討し、どちらを採用するかを決定しなければならない。一方、提案手法では消費エネルギーの平均値を根拠として優劣を機械的に判断できる。例では CP 集合 C の評価値は 9.5、CP 集合 D の評価値は 10.9 となり、CP 集合 C の方が優れていると判断できる。

4. 提案手法を用いたタスク内解析・最適化の改善

本節では，提案した DEPS プロファイルの評価法を用いることでタスク内解析・最適化の改善につながった例を示す.

4.1 チェックポイント選定

本節では，提案手法を用いてより低消費エネルギーとなるチェックポイントの位置を選定する手法を示す．図 4.1 にチェックポイント選定のアルゴリズムを示す．チェックポイントの候補となる場所はプログラム中に多く存在し，最適なチェックポイントの位置を探索することは困難である．そこで図 4.1 のチェックポイント選定アルゴリズムは，グリーディ法に基づいてエネルギー削減効率の高いチェックポイントの探索を行う．選定する CP の最大

Input: 選定する CP の最大数 $maxcp$, CP 候補リスト $cpcand$, CP 間見積もりデータ
Output: 確定 CP リスト $fixedcp$

1: **while** $fixedcp$ のサイズ $< maxcp$ **do**
2: $min\_score = 0$
3: **for all** $i \in cpcand$ **do**
4: $fixedcp + i$ の DEPS プロファイル生成
5: $score_i \leftarrow$ DEPS プロファイルの評価値
6: **if** $min\_score > score_i$ or $min\_score = 0$ **then**
7: $min\_score \leftarrow score_i$
8: $bestcp \leftarrow i$
9: **end if**
10: **end for**
11: $fixedcp \leftarrow fixedcp + bestcp$
12: $cpcand \leftarrow cpcand - bestcp$
13: $prev\_score \leftarrow min\_score$
14: **end while**
15: **return** $fixedcp$

図 4 チェックポイント選定アルゴリズム

数 $maxcp$ と CP 候補リスト $cpcand$ を入力とし，有効なチェックポイントリストを出力する．また，$cpcand$ の全ての候補が有効な場合のチェックポイント間の実行時間と消費エネルギーの見積もりが得られているものとする．確定済みのチェックポイントとチェックポイント候補の中のひとつを有効にした DEPS プロファイルを生成し，DEPS プロファイルの評価を行う (4 行目, 5 行目)．これを全てのチェックポイント候補に対して実行し，その中で最も評価値が低いチェックポイント候補を確定チェックポイントリストに加える．以上を確定チェックポイント数が $maxcp$ に達するまで繰り返す.

4.2 通過順序を考慮したチェックポイントの動作

本節では，通過順序を考慮したチェックポイントの動作について述べる．まず，チェックポイントを挿入することで実行時間，消費エネルギーの特性が悪化する場合について述べる．あるテストデータと別のテストデータでチェックポイントを通過する順序が逆になる場

Vol.2011-SLDM-149 No.4
Vol.2011-EMB-20 No.4
2011/3/18

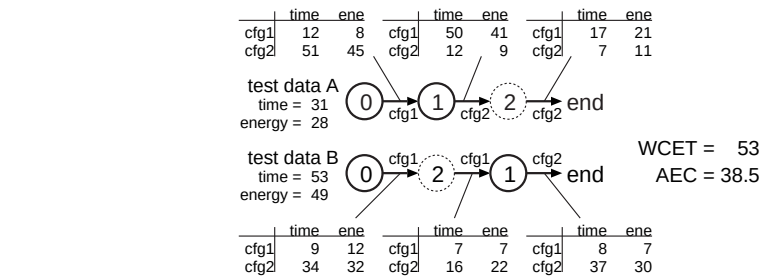
合がある．チェックポイントの通過順序が逆転するような箇所にチェックポイントを挿入すると，タスクの実行時間，消費エネルギーが悪化する可能性がある．図 5 に，チェックポイントを挿入することで性能が悪化する例を示す．図 5(a) では CP2 は挿入されていないため，破線の丸で示された CP2 の前後ではコンフィグレーションは変更されない．図 5(b) では CP2 が挿入されており，図では cfg1 を選択している．テストデータ A の CP2 で cfg1 を選択することは効率が悪く，テストデータ A の消費エネルギーが大きくなっている．CP2 で cfg2 を選択する場合はテストデータ B で性能が低下するため，どちらを選択しても CP2 を挿入することで全体の性能が低下することとなる．

チェックポイントの通過順序の逆転を解消するために，チェックポイントの通過履歴を考慮することを考える．これまでに通過してきたチェックポイントによってこれから通過するチェックポイントの動作を変更する．ただし，多くの履歴を考慮すると実行時のチェックポイント処理が複雑になり，チェックポイントのオーバヘッドが増大する．そこで，本稿ではチェックポイントに通過すべき順序を設定し，その通過順序を守っていないチェックポイントを無効にすることを考える．このとき，途中に通過しないチェックポイントがあっても構わない．この場合，チェックポイント処理では直前に通ったチェックポイントの ID のみを記録しておき，次に通過するチェックポイントの ID と照合することで，通過順序が守られているかどうかを調べることができる．図 5(c) に通過順序を考慮したチェックポイントの動作例を示す．ここでは，CP0→CP1→CP2 を規定の通過順序としている．テストデータ B では，CP2 の後に CP1 を通過しており，CP1 が無効となりコンフィグレーションが CP1 前後で変化しない．

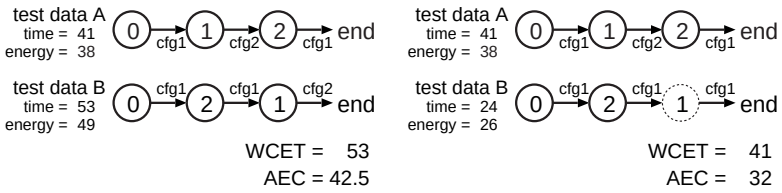
4.3 評　　価

アプリケーションの動作環境として，MeP[6] をベースとし，動作周波数を 2 通り，キャッシュウェイ数を 4 通り，全体で 8 通りのハードウェア構成をとることができるプロセッサ[7] を想定する．評価にはビデオのエンコードタスク (VENC) とデコードタスク (VDEC) を用いた．DEPS プロファイル生成の入力となるチェックポイント間の消費エネルギーと実行時間は，MeP 用シミュレータで取得した実行トレースを用いて見積もった．実行時間は実行トレースのサイクル数を見積もりとして用い，消費エネルギーは 8) の手法に基づくシミュレータを用いて見積もった．DEPS プロファイル生成アルゴリズムを perl で実装し，Xeon X5680 3.33GHz, 65GB memory, CentOS 5.5 上で実行した．

チェックポイントを DVFS に基づく手法[3] で選定した場合，提案する DEPS プロファイルの評価法を用いて選定した場合，さらに通過順序を考慮して選定した場合の 3 通りを比較



(a) CP1 のみの場合



(b) CP2 を挿入した場合　　(c) 通過順序を考慮した場合

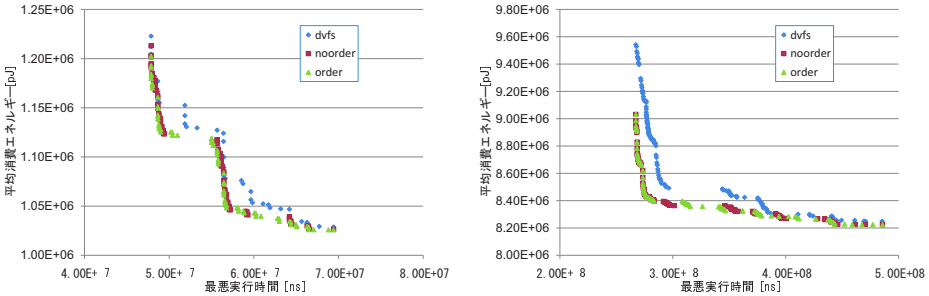
図 5　チェックポイントの通過順序と性能

した．DVFS に基づく手法は実行サイクル数と消費エネルギーの関係をモデル化し，その関係から消費エネルギーを最小化するようにチェックポイントの位置を決定しており，本稿で提案した評価手法を用いていない．DVFS に基づく手法では，VDEC，VENC ともに 8 個のチェックポイントを設定した．提案手法を用いたチェックポイント選定では，まず DVFS に基づく手法によって 60 個のチェックポイント候補を用意し，その中から VENC は 8 個，VDEC は 7 個のチェックポイントを選定する．チェックポイントの通過順序を考慮する場合は，各チェックポイントにおける残り最悪実行時間の多い順を守るべき通過順序とした．これらの手法で選定した位置にチェックポイントを挿入し，DEPS プロファイルを生成，評価した．表 1 に DEPS プロファイルの評価値とチェックポイント選定に要した時間を示す．DVFS に基づく手法に比べ，提案手法を用いた 2 つは評価値が改善されていることがわかる．また，VDEC では通過順序を考慮すると，より評価値が小さくなっている．図 6 に得られた DEPS プロファイルの最悪実行時間，平均消費エネルギーをプロットしたグラフを示す．VDEC，VENC ともに DVFS に基づく手法よりも明らかに改善されている．また，

Vol.2011-SLDM-149 No.4
Vol.2011-EMB-20 No.4
2011/3/18

表 1 DEPS プロファイルの定量的評価

	VDEC			VENC		
	評価値	改善率	選定時間	評価値	改善率	選定時間
DVFS に基づく手法	1088066	-	-	8434231	-	-
提案手法	1073534	1.34%	396min	8315315	1.41%	2359min
提案手法+通過順序	1071539	1.52%	351min	8315545	1.41%	2347min



(a) VDEC (b) VENC

図 6 DEPS プロファイルの視覚的評価

VDEC では通過順序を考慮した方がわずかに優れており，提案手法を用いた定量的評価と一致する．VENC ではチェックポイントの通過順序を考慮しない場合，した場合の優劣は視覚的には判断しづらい．表 1 の定量的評価においても評価値の差はわずかである．以上より，提案した DEPS プロファイルの評価法は視覚的な判断とよく一致しており，機械的に DEPS プロファイルの優劣を判別するために有用であるといえる．

5. ま と め

本稿では DEPS プロファイルの定量的評価法を提案した．提案手法では DEPS プロファイルからデッドラインついての消費エネルギーの平均値を算出し，評価値とした．提案手法の有用性を示すために，チェックポイント選定，通過順序を考慮したチェックポイント処理の二つの例を示した．提案手法を用いることで，DVFS に基づく手法と比較して優れた DEPS プロファイルが得られる位置にチェックポイントを設定することができた．また，DEPS プロファイルをプロットしたものと提案手法による評価値を比較したところ，視覚的な評価ともよく一致しており，提案手法は機械的に DEPS プロファイルの優劣を判断するのに有用であることを示すことができた．

提案手法では評価値として，デッドラインが一様に分布していると仮定して平均値を求めている．DEPS プロファイルの評価法の課題として，他のデッドライン分布の検討が挙げられる．チェックポイント選定については，選定の過程で多くの DEPS プロファイルを生成する必要がありそれぞれの DEPS プロファイル生成に時間がかかっているため，より多数のチェックポイントを選定するのが困難である．[4] では全数探索の効率化を行っているが，高速に DEPS プロファイルを生成するために近似的に探索する手法が必要である．

謝　　辞

本研究は，科学技術振興事業団 (JST) 戦略的創造研究推進事業 (CREST)「情報システムの超低消費電力化を目指した技術革新と統合化技術」の支援による．

参 考 文 献

1) G. Zeng, et al., “A Generalized Framework for Energy Savings in Hard Real-Time Embedded Systems,” IPSJ Transactions on System LSI Design Methodology, Vol.2, pp.167-179, Aug. 2009.
2) 高田広章, “ソフトウェアとハードウェアの協調による組込みシステムの消費エネルギー最適化”, 情報処理, Vol.51, No.7, July 2010.
3) T. Tatematsu, et al., “Checkpoint Extraction Using Execution Traces for Intra-Task DVFS in Embedded Systems,” Proc. of DELTA 2011, Queenstown, New Zealand, Jan. 2011.
4) 川島裕崇, 曾剛, 渥美紀寿, 立松知紘, 高瀬英希, 高田広章, “DEPS フレームワークにおける平均消費エネルギーと最悪実行時間のタスク内解析手法,” 情報通信学会技術報告, Mar. 2011 (発表予定).
5) H. Takase, et al., “Partitioning and allocation of scratch-pad memory for priority-based preemptive multi-task systems,” Proc. of DATE 2010, pp.1124-1129, Mar. 2010.
6) 株式会社東芝, MeP, http://www.semicon.toshiba.co.jp/product/micro/selection/mep/index.html
7) T. Ishihara, et al., “AMPLE: An Adaptive Multi-Performance Processor for Low-Energy Embedded Applications,” Symposium on Application Specific Processors, pp.83-88, Jun. 2008.
8) D. Lee, et al., “An Energy Characterization Framework for Software-Based Embedded Systems,” ESTIMedia2006, Vol.1, pp.59-64, Oct. 2006.