Energy Optimization of Stochastic Applications with Statistical Guarantees of Deadline and Reliability

Xiong Pan¹, Wei Jiang¹, Ke Jiang², Liang Wen¹, and Qi Dong¹

¹School of Information and Software Engineering
University of Electronic Science and Technology of China

²Department of Computer and Information Science, Linköping University, Sweden
Outline of this paper

- Introduction
- System model
- System problem formulation
- Proposed approaches based on dynamic programming
- Experimental results
- Conclusions
Introduction

Paper focus on:

- Optimizing energy consumption of stochastic tasks;
- Take into account the probability distribution of task’s execution time;
- The application has reliability and deadline constraints;
- Running on DVFS-enabled embedded systems.
System Model

Stochastic real-time task model:

Scenario and Assumption

- An embedded server to provide real-time services for multiple users in a common period.
- All service tasks are released in each period, and finished before their respective deadlines.

We use \((D_i, C_i, PDF_i)\) to characterize the task \(i\).

- \(D_i\): relative deadline;
- \(C_i\): stochastic execution time;
- \(PDF_i\): probability density function of \(C_i\).
System Model

**DVFS-enabled energy and reliability model:**

- Expected energy consumption of task $i$:

  $$\overline{En_i} = (\text{Pow}_i^{\text{dyn}} + \text{Pow}_i^{\text{leak}}) \cdot \overline{C_i}$$

- Reliability of task $i$:

  $$Re_i = \exp\left(-\frac{\lambda(\phi_i) \cdot \overline{C_i}}{f_i}\right)$$

- Expected logarithm of the reliability of the whole application can be calculated as:

  $$\overline{\ln(ORE)} = \sum_{i=1}^{n} \left(-\frac{\lambda(\phi_i) \cdot \overline{C_i}}{f_i}\right)$$
We need $\ln(ORe)$ or $\overline{ORe}$ for calculating the reliability, but we only have got $\overline{\ln(ORe)}$, so formulating the following theorem:

**Theorem 1** The natural logarithm of expected system reliability can be expressed as $\ln(ORe) = \ln(ORe)$, approximately.

The theorem can be proved by Taylor expansion and Taylor Mean Value Theorem.
Modeling the Probability of No-Deadline Violation

To model the deadline constraint of soft real-time applications, we introduce a metric, Probability of No-Deadline Violation (PNDV), indicating that the deadline violation for each task can only be tolerated by certain probability.

- $Pb_i^h$ denotes the probability of the situation when the actual execution time of $\tau_i$ is $h$;
- $OPb_i^h$ denotes the probability that the overall execution duration of the previous $i$ tasks is $h$;
- $Te_i$ denotes the sum of the execution durations of the previous $i$ tasks.
System Model

Modeling the Probability of No-Deadline Violation

Sum of the execution durations of the previous $i$ tasks:

$$ Te_i = \sum_{j=1}^{i} C_j. $$

Given the probability distribution of $Te_{i-1}$, Probability distribution of $Te_i$ can be calculated easily by:

$$ OPb_i^h = \sum_{x+y=h} (OPb_{i-1}^x \cdot Pb_i^y) $$

At last, $PNDV$ of task $i$ can be expressed as follows:

$$ PNDV_i = \sum_{h=1}^{D_i} OPb_i^h $$
The problem we address is described as follows. Given a task set with $n$ tasks $T_1, T_2, \ldots, T_n$, with corresponding deadline, execution time PDF and $m$ possible levels of processor frequency, we are interested to determine a DVFS solution, i.e. $S = (f_1, f_2, \ldots, f_n)$, such that expected system energy for a period of the tasks is minimized and the constraints of the reliability ($OR_{\bar{B}}$) and deadlines ($PNDV_{\bar{B}}$) are met.

The optimization objective can be formulated as:

$$\text{Minimize} \quad OEn = \sum_{i=1}^{n} (Pow_{i}^{\text{dyn}} + Pow_{i}^{\text{leak}}) \cdot E_{pt}(C_i)$$
Soft deadline constraint of individual task can be expressed:

\[ \text{PNDV}_i \geq \text{PNDV}_i^B, 1 \leq i \leq n \]

Reliability constraint can be expressed:

\[ \sum_{i=1}^{n} \left( -\frac{\lambda(\phi_i) \cdot \overline{C_i}}{f_i} \right) \geq \ln(\text{ORe}^B), f_i \in \{F_1, F_2, \ldots, F_m\} \]

\{F_1, F_2, \ldots, F_m\} are the normalized discrete frequency options for the DVFS-enabled processor.
Proposed Approaches

**DPO approach** *(Dynamic Programming based Optimization algorithm)*

It is a Multidimensional Multiple-Choice Knapsack Problem.

The state transition equation can be expressed as following:

\[
\Phi(i, \delta, k) = \max_k \{ \Phi(i, \delta - 1, l), \\
\max \{ \Phi(i - 1, \delta - En_i^j, l) + \ln Re_i^j \} \} \\
1 \leq j \leq m, 1 \leq l \leq k
\]

- \( \Phi(i, \delta, k) \): expected natural logarithm of the k-th largest reliability of the previous i tasks, whose expected total energy is limited to no more than \( \delta \).
- \( En_i^j \) and \( Re_i^j \): energy and reliability of \( \tau_i \) using the j-th frequency level.
Proposed Approaches

**FBDP approach**
*(Fast Bi-search algorithm based on Dynamic Programming)*

- Due to high runtime complexity of DPO, we propose FBDP.
- FBDP is to find a solution satisfying both reliability and deadline constraints with the energy cost at most \((1+\beta)OEn^*\), where \(OEn^*\) denotes the expected energy of our optimal problem.
- Achieving this, a Bi-searcher will be first constructed such that a bi-search on whether \(OEn^* \geq x\) for any \(x\) can be found in polynomial time without knowing the value of \(OEn^*\). We can use Bi-searcher to bi-narrow the range of \(OEn^*\).
- After constructing such a Bi-searcher, the range of \(OEn^*\), which is narrow enough, can be obtained by performing a fast bi-search between the lower and upper bounds of \(OEn^*\).
Proposed Approaches

FBDP approach

Construction of Bi-searcher:

Scale $\tilde{E}n_i^j$ to $\left[\tilde{E}n_i^j \cdot n/(x\beta)\right]$ and reliability and deadline constraints are unchanged. The upper bound of expected energy cost is set to be $n/\beta$, after executing DPO and we can determine the range of $OEn^*$ based on the following 2 cases:

Case 1: Found a solution satisfying both reliability and deadline constraints. Then $OEn^* \leq (1+\beta)x$.

Case 2: No solution is found that satisfied both reliability and deadline constraints. Then $OEn^* > x$. 
Proposed Approaches

**FBDP approach**

After each bi-search, we can make corresponding operations according to the previous two cases:

**Case 1:** $OEn^* \leq (1+\beta)x$. Set new upper bound to $(1+\beta)x = \sqrt{(OEn^*_l \cdot OEn^*_u) / (1 + \beta)}$ and keep lower bound unchanged.

**Case 2:** $OEn^* > x$. Set new lower bound to $\sqrt{(OEn^*_l \cdot OEn^*_u) / (1 + \beta)}$ and keep upper bound unchanged.

It can be proved that the ratio of upper bound to lower bound is no more than $2(1+\beta)$ after $\log(\log Q_0)$ bi-search operations, and obtain the near-optimal solution with the energy cost at most $(1+\beta)OEn^*$ without violating the reliability and deadline constraints.
Experimental Results

Parameters on tasks in all the experiments are generated randomly:
- Deadline -- [30 ms, 2000 ms]
- WCET -- [20 ms, 80 ms]
- PDF -- [Gaussian Distribution, Uniform Distribution or Exponential Distribution]
- Expected values of tasks’ execution durations -- [5, 30]
- variances of tasks’ execution durations -- [0.2, 2]

Comparing algorithms
- WCET-OPT -- takes the WCET as the actual execution time of each task;
- NO-DVFS -- ignores DVFS capability of the processor;
- DPO -- Dynamic Programming based Optimization algorithm;
- FBDP -- Fast Bi-search algorithm based on Dynamic Programming.
Experimental Results

Impact of PNDV constrain:

(a)

(b)
Experimental Results

Impact of PNDV constrain:

(c) Average PNDV vs. PNDV constraint
(d) Execution time (ms) vs. PNDV constraint
Experimental Results

Impact of task number:

![Graph showing normalized expected energy cost and expected reliability vs task number for different algorithms.](image)
Experimental Results

Impact of task number:

![Graph showing impact of task number on PNDV and execution time.]

- Average PNDV
- Execution time (ms)
- Graphs compare DPO, FBDP(β=0.05), FBDP(β=0.2), NO-DVFS, and WCET-OPT.
Experimental Results

A real-life case:

We consider an embedded server like satellite server or radar server, which has to provide services for 30 users within a common period of 3000 ms. The task for each user all arrives at the same time and has to be finished before the next period.

<table>
<thead>
<tr>
<th>Constraint</th>
<th>$O_E$</th>
<th>$N_{RV}$</th>
<th>$N_{ND}$</th>
<th>Time (ms)</th>
<th>$O_E$</th>
<th>$N_{RV}$</th>
<th>$N_{ND}$</th>
<th>Time (ms)</th>
<th>$O_E$</th>
<th>$N_{RV}$</th>
<th>$N_{ND}$</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.91</td>
<td>0.71536</td>
<td>9926</td>
<td>289456</td>
<td>6078</td>
<td>0.74269</td>
<td>9930</td>
<td>290386</td>
<td>148</td>
<td>1</td>
<td>10000</td>
<td>35510</td>
<td></td>
</tr>
<tr>
<td>0.93</td>
<td>0.71536</td>
<td>9926</td>
<td>289456</td>
<td>8063</td>
<td>0.75317</td>
<td>9951</td>
<td>291323</td>
<td>197</td>
<td>1</td>
<td>10000</td>
<td>35712</td>
<td></td>
</tr>
<tr>
<td>0.95</td>
<td>0.76451</td>
<td>9949</td>
<td>292528</td>
<td>6047</td>
<td>0.78376</td>
<td>9973</td>
<td>295008</td>
<td>149</td>
<td>1</td>
<td>10000</td>
<td>35946</td>
<td></td>
</tr>
<tr>
<td>0.97</td>
<td>0.84789</td>
<td>9971</td>
<td>295984</td>
<td>7063</td>
<td>0.87254</td>
<td>9979</td>
<td>296121</td>
<td>163</td>
<td>1</td>
<td>10000</td>
<td>36025</td>
<td></td>
</tr>
<tr>
<td>0.99</td>
<td>0.92214</td>
<td>9987</td>
<td>298416</td>
<td>6062</td>
<td>0.93998</td>
<td>9989</td>
<td>299367</td>
<td>141</td>
<td>1</td>
<td>10000</td>
<td>36213</td>
<td></td>
</tr>
</tbody>
</table>
Conclusions

- Consider PDFs of tasks’ execution into the design problem;
- DVFS is used for energy saving;
- Design FBDP which can achieve approximated optimal solution;
- Experiment results illustrated the effectiveness of proposed technique;
- used Monte Carlo simulation to simulate the stochastic tasks for verifying the proposed approaches.
That is all my presentation
Thanks for your time