Quality-aware Video Decoding on Thermally-constrained MPSoC Platforms

Deepak Gangadharan, Samarjit Chakraborty and Jürgen Teich
Chair of Hardware-Software Co-Design, Friedrich-Alexander-Universität Erlangen-Nürnberg
Institute for Real-Time Computer Systems, TU Munich
Motivation

- Higher power consumption on PEs due to technology scaling $\rightarrow$ higher power densities $\rightarrow$ undesirable hot spots (localized high temperature points)
- Processor intensive multimedia workloads !!
- Traditional solution: Expensive cooling packages – Not viable for cost-constrained portable devices
- Widely accepted solution: Dynamic Thermal Management (DTM)
Motivation (contd…)

- DTM techniques for video decoding that exist in literature trade off quality for safe thermal profiles

- Video quality is important for applications where user experience of the video content is an important performance objective

- All previous DTM techniques for video decoding keep the thermal profile below a maximum peak temperature with no formal technique to bound quality degradations

- Our approach: A combined offline-online framework for video decoding that satisfies thermal constraints while also ensuring bounds on video quality
Background

- Two types of DTM techniques widely employed
  i) Dynamic Voltage/Frequency Scaling (DVFS) – Needs multiple voltage/frequency settings
  ii) Dynamic Power Management (DPM) – Consists of operation modes and system can be put into low power modes

In this work, we use DPM to adhere to thermal constraints
High-level overview

Higher inserted idle times without frame drops

Lower inserted idle times with frame drops

Deepak Gangadharan et al, ASAP 2014
Problem Description

Given: platform parameters, a QoS model, a thermal model, the target quality constraint, temperature

Goal: Compute the minimal idle times with appropriate frame drops such that and are adhered to
Schematic Overview

Offline Process

- Input stream
- QoS model
- $Q^I(f,L)$
- Drop Pattern Generation
- Frame drop pattern

Online Process

- Thermal model
- Idle times with frame drops such that $Q^I(f,L) >= Q_{target}$ and peak temperature $<= T_{max}$
Computation of

Video clip frame sequence

For red, green and blue pixel components, compute

Maximum Mean square error (MSE_{max}) = \sum_w \sum_h (p_d(h,w) - p_c(h,w))^2

Pixel intensity of Dropped frame

Pixel intensity of Concealment frame

Why MSE_{max}? \rightarrow Quality \quad (1/MSE) \rightarrow Gives the worst-case quality
Computation of

\[
\begin{align*}
&\text{I P B B P B B P B B P …} \\
&\text{i i + F}
\end{align*}
\]

Record the maximum MSE in the entire clip if ‘f’ frames are dropped in an interval of ‘F’ frames. Here 1 \( f \) \( F \).

\[Q^i(F) \text{ (in PSNR)} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}_{\text{maxnorm}}} \right)\]

Here \( \text{MSE}_{\text{maxnorm}} \rightarrow \) Maximum MSE normalized over the number of dropped frames \( f \) in the interval \( F \)
Thermal Model

**Lumped RC Model**

\[ \text{Steady state temperature} \]

- \( P \) – Average power dissipated over time \( t \) (in W)
- \( R \) – Thermal Resistance (°C/W)
- \( C \) – Thermal Capacitance (in J/°C)

Initial Temperature of the PE

Two modes of operation of the PE – Active mode ( ) and Idle mode ( )

Deepak Gangadharan et al, ASAP 2014
Workload curve and Workload class

– Workload cycles required to process first frames

\[
\begin{aligned}
M & \\
M + n & \\
\end{aligned}
\]

Record the max in this window

Workload Class: A workload class is a group of video clips for which the workload cycles for consecutive frames is upper bounded by
Quality and Thermal-aware Idle Time Insertion

1) From the drop pattern, determine the active run $k$, where active run is the sequence of frames that cannot be dropped

$$T_a + (T_{idle\_end} - T_a) e^{-\frac{(y^d(k)+OC) \times T_p}{RC}} \leq T_{max}$$

$$\iff T_{idle\_end} \leq T_a + (T_{max} - T_a) e^{-\frac{(y^d(k)+OC) \times T_p}{RC}}$$

- Dropped frame
- Active frame

2) Compute the safe temperature $T_{idle\_end}$ at the end of idle period

Required $T_{idle\_end}$ after idle time insertion

- Maximum no. of processor cycles for $k$ consecutive frames,
$OC$ – Overhead of the algorithm
Quality and Thermal-aware Idle Time Insertion

3) Compute idle time with being the temperature after the dropped frame

\[ T_i + (T_{prev} - T_i)e^{-\frac{t_{idle}}{RC}} \leq T_{idle\_end} \]

\[ \Leftrightarrow t_{idle} \geq R \times C \times \log\left( \frac{T_{prev} - T_i}{T_{idle\_end} - T_i} \right) \]

4) Temperature is updated as we traverse through the stream

5) Workload history is also updated
Importance of Workload History

**Theorem**

If \((\cdot)\) is the workload for \(k\) consecutive frames with online updates and \((\cdot)\) for \(k\) consecutive frames without history, then accumulated idle time inserted with \((\cdot)\) is always lesser than or equal to the accumulated idle time inserted with \((\cdot)\)
Case Study 1: MPEG-2 Decoder

1)

2) Five 8 Mbps video clips used, Tasks mapped on – VLD and IQ

3) $C = 112.2 \text{ mJ}, R = 1.83 \text{ /W}, \text{OC} = 3000 \text{ cycles}$

4) $= 90$, , , ,
## Reduction in idle times

<table>
<thead>
<tr>
<th>Clip</th>
<th>PE1 Delay (in sec)</th>
<th>HIST_MAX</th>
<th>0</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>susi_080 (QC1)</td>
<td>30</td>
<td>0.0917</td>
<td>0.0893</td>
<td>0.0891</td>
<td>0.0889</td>
<td>0.0889</td>
<td>0.0889</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>0.6374</td>
<td>0.6309</td>
<td>0.6298</td>
<td>0.6294</td>
<td>0.6277</td>
<td>0.6277</td>
<td></td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>0.5007</td>
<td>0.4944</td>
<td>0.4936</td>
<td>0.493</td>
<td>0.4921</td>
<td>0.4914</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>1.4958</td>
<td>1.4894</td>
<td>1.4891</td>
<td>1.4888</td>
<td>1.4857</td>
<td>1.4854</td>
<td></td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>cact_080 (QC2)</td>
<td>30</td>
<td>0.7998</td>
<td>0.7998</td>
<td>0.7917</td>
<td>0.7911</td>
<td>0.7911</td>
<td>0.7911</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>1.3286</td>
<td>1.3227</td>
<td>1.3107</td>
<td>1.3095</td>
<td>1.3093</td>
<td>1.3093</td>
<td></td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>5.832</td>
<td>5.6976</td>
<td>5.6976</td>
<td>5.6976</td>
<td>5.6976</td>
<td>5.6976</td>
<td></td>
</tr>
<tr>
<td>flwr_080 (QC3)</td>
<td>30</td>
<td>1.0101</td>
<td>0.9974</td>
<td>0.99</td>
<td>0.9896</td>
<td>0.9896</td>
<td>0.9896</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>2.7522</td>
<td>2.7522</td>
<td>2.7466</td>
<td>2.7466</td>
<td>2.7466</td>
<td>2.7466</td>
<td></td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>mobl_080 (QC2)</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>0.0679</td>
<td>0.0626</td>
<td>0.0626</td>
<td>0.0626</td>
<td>0.0626</td>
<td>0.0626</td>
<td></td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>0.7183</td>
<td>0.7183</td>
<td>0.7183</td>
<td>0.711</td>
<td>0.711</td>
<td>0.711</td>
<td></td>
</tr>
<tr>
<td>orion_2 (QC2)</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>0.0679</td>
<td>0.0626</td>
<td>0.0626</td>
<td>0.0626</td>
<td>0.0626</td>
<td>0.0626</td>
<td></td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>0.7183</td>
<td>0.7183</td>
<td>0.7183</td>
<td>0.711</td>
<td>0.711</td>
<td>0.711</td>
<td></td>
</tr>
</tbody>
</table>

1) Reduction in idle times with small decrease in quality (2 dB) – *mobl_080* (1.7 s)

2) Reduction with history – *mobl_080* (20 ms)
Case Study 2: H.264 decoder

1) Five 3.25 Mbps video clips used, Tasks mapped on – CABAC and Deblocking filter
2) RxC = 0.4326 , OC = 3000 cycles
3) = 92 , , ,
Reduction in idle times

<table>
<thead>
<tr>
<th>PE Delay (in sec)</th>
<th>HIST_MAX</th>
<th>0</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>clip</td>
<td>PSNR (in dB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>1.1269</td>
<td>1.124</td>
<td>1.1153</td>
<td>1.1153</td>
<td>1.1116</td>
<td>1.1112</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>1.1269</td>
<td>1.124</td>
<td>1.1153</td>
<td>1.1153</td>
<td>1.1116</td>
<td>1.1112</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>2.4277</td>
<td>2.426</td>
<td>2.4241</td>
<td>2.4238</td>
<td>2.4189</td>
<td>2.4186</td>
</tr>
<tr>
<td>nVidia</td>
<td>20</td>
<td>7.8066</td>
<td>7.8007</td>
<td>7.7862</td>
<td>7.7862</td>
<td>7.7862</td>
<td>7.7862</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>&gt; 10</td>
<td>&gt; 10</td>
<td>&gt; 10</td>
<td>&gt; 10</td>
<td>&gt; 10</td>
<td>&gt; 10</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>&gt; 10</td>
<td>&gt; 10</td>
<td>&gt; 10</td>
<td>&gt; 10</td>
<td>&gt; 10</td>
<td>&gt; 10</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>5.2107</td>
<td>5.2053</td>
<td>5.1844</td>
<td>5.1799</td>
<td>5.1735</td>
<td>5.1722</td>
</tr>
</tbody>
</table>

1) Reduction in Idle time with small decrease in quality (2 dB) – *ultimatum* (3.0168 s)

2) Reduction with history – *ultimatum* (27 ms)
Concluding Remarks

- Proposed a combined offline-online approach to decode video streams in quality and thermal-aware manner

- Inserted idle times are reduced combining application level technique of frame drops under target quality constraints with maintenance of workload histories

- **Future work**: Explore the possible optimizations to reduce the idle times and end-to-end delays while jointly considering neighboring PEs
Thank You!!

Questions??