Efficient Multimedia Streaming for Power Aware Devices: A survey

Gautam Gopinadhan
NCSU
Outline

- Introduction
- Categorization of Solutions
  - Category 1: DVS Techniques
  - Category 2: Network level optimizations
  - Category 3: Content Adaptation
  - Category 4: Architecture Changes
- Conclusion
Introduction

- What are Power Aware Devices? What are their characteristics?
  - Running on battery power – energy consumption is crucial
    • Laptops, handhelds, mobile phones
  - Usually have Power Saving Features:
    • Power saving modes
    • Ability to switch off certain components (eg. Caches, memory banks)

<table>
<thead>
<tr>
<th>Frequency</th>
<th>300</th>
<th>500</th>
<th>600</th>
<th>700</th>
<th>800</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage</td>
<td>1.2</td>
<td>1.2</td>
<td>1.25</td>
<td>1.3</td>
<td>1.35</td>
<td>1.4</td>
</tr>
<tr>
<td>Relative Power (%)</td>
<td>22.04</td>
<td>36.73</td>
<td>47.83</td>
<td>60.35</td>
<td>74.39</td>
<td>100</td>
</tr>
</tbody>
</table>
Introduction

- Characteristics of Multimedia tasks
  - High consumption of CPU
  - High consumption of Networking resources (streaming)
  - High consumption of display resources (video)
  - Real time characteristics
  - Demands for resources varies a lot – depends on content and media application
  - Low spatial locality in media data (poor caching behaviour)

- Research challenge to optimize energy consumption!
Introduction: MPEG basics

- MPEG video content -> sequence of images or “frames”
- 3 types of frames: I, P, B
  - I frame -> Intra frame (medium compression)
    - I frame is independently coded
  - P frame -> Predicted frame (higher compression)
    - P frame differentially coded with respect to a previous I frame or a P frame
  - B frame -> Bidirectional frame (high compression)
    - B frame differentially coded with respect to a I/P frame ahead and I/P frame behind
Introduction: MPEG basics

- General Characteristics of various MPEG frames
  - Decoding expense: I > B > P
  - Decoding expense for 2 frames of the same type is usually similar in the short term
  - In the long term there can be high variability in the decoding times for a particular frame type

Power consumption of video decoding:

![Graph showing power consumption over video frame sequence number]

*Figure 1: Power consumption of video decoding (15 fps).*
Categorization of solutions

- Main source of energy consumption
  - CPU/memory + additional circuit boards (~1-3 W)
  - Network hardware (1.4 W)
  - Display (~1 W)

- Solutions mainly target these specific sources.
Categorization of Solutions

- Solutions categorized into following
  - DVS Techniques
    - DVS guided by CPU utilization prediction
  - Network level optimizations
    - Network device aware delivery optimizations
  - Content adaptation
    - Adapting media content to decrease processing
  - Architecture changes
    - Reconfiguration of hardware units (Simulated)
Category 1: DVS Techniques

- **What is DVS?**
  - Most mobile processors have Dynamic Voltage Scaling (DVS) support.

  - **What is DVS?**
    - Power = $C \times f \times V^2$
    - $C$ = Circuit capacitance
    - $F$ = Frequency of Processor
    - $V$ = Supply voltage
    - Reducing voltage greatly reduces power
    - **Concern: Reducing Voltage also reduces $f$.**
    - Ensure deadline can be met under this low frequency

  - **Spread “slack time” by reducing frequency of processor**
DVS Techniques

- General technique:
  - Try to predict CPU requirement for decoding a media unit
  - Switch CPU frequency to appropriate level so media unit can be decoded.

- Pouwelse et al

- Insight: Strong correlation between frame size and decoding time

![Figure 3: Decoding times for P frames @ 236 MHz.](image)
DVS Techniques

- Pouwelse et al

Between 17% - 75% reduction
DVS Techniques

- Bavier et al.
- Average decoding time for previous frame of same frame type recorded
- Weighted adjustment depending on previous prediction success applied to get prediction value.

Result:
- Upto 80% improvement in power requirement
- About 10-20% deadline misses
Network Level Optimizations

- Network cards have various power consumption characteristics
  - Cisco Aironet 350 series
    - Transmit (1.68 W)
    - Receive (1.435 W)
    - Idle (1.34 W)
    - Sleep (0.184 W)

- Main idea:
  - Put card in sleep mode as often as possible
Network Level Optimizations

- Instead of sending data all the time, a Proxy sends data in bursts (Chandra et al)

- Burst send schedule is communicated with clients at scheduled times by proxy

- Clients receive data during scheduled times and sleep otherwise

- Result: Over 75% energy saved

![Network Diagram]
Content Adaptation

- Work by Cornea et al.

- Key insight:
  - Media can be transcoded with different parameters (frame rate, bit rate, resolution) without noticeable change to human perception

- Concern: Relationship between media encoding parameters and CPU requirement at client is not clear
  - Use empirical methods to correlate parameter values to power consumption on client
Content Adaptation: Cornea et al.

- **Discrete levels of video quality in terms of parameters**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Like Original (No improvement required)</td>
<td>SIF, 30fps, 650Kbps</td>
<td>4.42 W</td>
<td>6.07 W</td>
</tr>
<tr>
<td>Excellent</td>
<td>SIF, 25fps, 450Kbps</td>
<td>4.37 W</td>
<td>5.99 W</td>
</tr>
<tr>
<td>Very Good</td>
<td>SIF, 25fps, 350Kbps</td>
<td>4.31 W</td>
<td>5.86 W</td>
</tr>
<tr>
<td>Good</td>
<td>HSIF, 24fps, 350Kbps</td>
<td>4.24 W</td>
<td>5.81 W</td>
</tr>
<tr>
<td>Fair</td>
<td>HSIF, 24fps, 200Kbps</td>
<td>4.15 W</td>
<td>5.73 W</td>
</tr>
<tr>
<td>Poor</td>
<td>HSIF, 24fps, 150Kbps</td>
<td>4.06 W</td>
<td>5.63 W</td>
</tr>
<tr>
<td>Bad</td>
<td>QSIF, 20fps, 150Kbps</td>
<td>3.95 W</td>
<td>5.5 W</td>
</tr>
<tr>
<td>Terrible (poorer quality not acceptable)</td>
<td>QSIF, 20fps, 100Kbps</td>
<td>3.88 W</td>
<td>5.38 W</td>
</tr>
</tbody>
</table>

- **Given energy level at client, maintain best possible video quality for duration of stream**

- **Proxy transcodes media according to parameter values**
Architecture Changes

- **General Idea**
  - Assume architectural units are configurable
  - Switch off units if not required

Fig: CPU internal power distribution
Architecture Changes: Cache Reconfiguration

- Change data cache size and associativity [Cornea et. Al.]
  - Decreasing cache size -> lower cache power consumption
  - Concern: MPEG decoding exhibits poor locality -> greater cache-memory traffic.
  - Use extensive simulations to record “sweet spot” for a video quality level
  - Switch to optimal cache configuration for that video quality
  - 10-20% improvement
Summary

- Multiple ways of supporting power aware multimedia applications

4 categories of solutions:
- DVS Techniques (widely studied)
- Network level optimizations
- Content adaptation (relatively new and recently applied to the power domain)
- Architecture adaptation (completely done through simulations, requires very specific hardware support)