Dynamic Voltage Scheduling Using Adaptive Filtering of Workload Traces

Amit Sinha and Anantha Chandrakasan

Massachusetts Institute of Technology
Overview

- Introduction
  - Typical Workload Profile
  - DVS Basics

- Energy Workload Models
- Workload Prediction
  - Markov Processes
  - Various Algorithms
- Energy Performance Tradeoffs
- Results and Conclusions
Typical Processor Workload Profiles

![Graph showing processor utilization over time for Dialup Server, Fileservlet, and Workstation]
Dynamic Voltage Scaling

Fixed Power Supply

 ACTIVE IDLE

\[ E_{\text{FIXED}} = \frac{1}{2} C V_{\text{DD}}^2 \]

Variable Power Supply

 ACTIVE

\[ E_{\text{VARIABLE}} = \frac{1}{2} C (V_{\text{DD}}/2)^2 = E_{\text{FIXED}} / 4 \]

Normalized Workload

Normalized Energy

Fixed Supply

Variable Supply

Normalized Workload
Enabling Technology

- Variable frequency processors available
  - Transmeta’s Crusoe
    - LongRun Technology
  - AMD K6-2+
    - PowerNOW!
  - Mobile Pentium III
    - SpeedStep

- StrongARM SA-1100
  - 59MHz – 206MHz (0.8V – 1.5V)
Energy Workload Model

Energy vs. Workload

No Voltage Scaling
DVS with Converter Efficiency
Ideal DVS

[Dutnik97]

\[ E(r) = CV_0^2 T_s f_{\text{ref}} r \left[ \frac{V_l}{V_0} + \frac{r}{2} + \sqrt{r \frac{V_l}{V_0} + \left( \frac{r}{2} \right)^2} \right]^2 \]

\[ I(r) = I_{\text{ref}} r \left( \frac{V_0}{V_{\text{ref}}} \left[ \frac{V_l}{V_0} + \frac{r}{2} + \sqrt{r \frac{V_l}{V_0} + \left( \frac{r}{2} \right)^2} \right] \right) \]
Workload Prediction

Can be modelled as a Markov Process

- How to predict workload, w?
- How frequently processing rate, f(r), be updated

Sinha, VLSI '01
### Prediction Algorithms

#### Moving Average Workload (MAW)

- Formula: $h_n[k] = \frac{1}{N}$ \(\forall n, k\)
- Characteristics:
  - Simplest
  - Performance degradation with fast loads

#### Expected Workload State (EWS)

- Formula: $w[n + 1] = \mathbb{E}\{w[n + 1]\} = \sum_{j=0}^{L} w_j p_{ij}$
- Characteristics:
  - Probabilistic formulation
  - Transition matrix updated every slot

#### Exp. Weighted Average (EWA)

- Formula: $h_n[k] = a^{-k}$
- Characteristics:
  - Lower significance of older data
  - Event prediction context [Hwang97]

#### Least Mean Square (LMS)

- Formula: $h_{n+1}[k] = h_n[k] + \mu w_e[n]w[n-k]$ (with $\mu$ being the step size)
- Characteristics:
  - Adaptive filter, self-adjusting
  - Convergence issues

#### Predicted Workload

$$w_p[n + 1] = \sum_{k=0}^{N-1} h_n[k]w[n-k]$$

#### Previous Workloads

$\{w[n], 1\} = \sum_{i=1}^{N-1} \varepsilon_i - \sum_{i=1}^{N-1} m_i w[n] = 0$
Prediction Performance

- Best prediction with LMS and about 3 taps

- Averaged over different processors and times
- 1 sec update rate
- 1 hour processor utilization snapshots

Graph showing RMS Error vs. Filter Taps (N)

- EWA
- EWS
- MAW
- LMS

Less Taps: Noisy Prediction
More Taps: Excessive LPF
LMS Tracking of Workload

\[ N = 3 \]
\[ T = 10 \]
\[ \text{Levels} = 10 \]
\[ \mu = 0.1 \]
**Energy Performance Tradeoff**

- Averaging is energy efficient

\[
\frac{r_1^2 + r_2^2}{2} \geq \left( \frac{r_1 + r_2}{2} \right)^2 \Rightarrow E(r) \geq E(\bar{r})
\]

- Update time T depends on
  - Maximum allowed performance hit
  - DC/DC converter and frequency change overheads
### Performance Hit Metric

#### Performance Hit Function

\[
\phi(\Delta t) = \frac{\bar{W}_{\Delta t} - \bar{r}_{\Delta t}}{\bar{r}_{\Delta t}}
\]

#### Maximum and Average

\[
\phi_{\max}(\Delta t), \phi_{\avg}(\Delta t)
\]

Maximum can be used set update time

---

- Performance Hit Function
- Maximum and Average

---

Sinha, VLSI '01
Optimum Update Time and Taps

- Good choice
  \( N = 3 \)
  \( T = 5 \) s

- \( N, T \) selections are not completely independent!
Discrete frequency levels are not too bad.

- StrongARM has 11 levels [ degradation < 5% ]
## Results

<table>
<thead>
<tr>
<th>Trace</th>
<th>Filter</th>
<th>Energy Savings Ratio (ESR)</th>
<th>ESR Comparison</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Max / Perfect</td>
<td>Max / Actual</td>
</tr>
<tr>
<td>Dialup Server</td>
<td>MAW</td>
<td>2.9</td>
<td>2.2</td>
<td>1.10</td>
</tr>
<tr>
<td></td>
<td>EWS</td>
<td>2.4</td>
<td>2.1</td>
<td>1.11</td>
</tr>
<tr>
<td></td>
<td>EWA</td>
<td>2.2</td>
<td>1.2</td>
<td>1.09</td>
</tr>
<tr>
<td></td>
<td>LMS</td>
<td>2.3</td>
<td>1.03</td>
<td>14.7</td>
</tr>
<tr>
<td>File Server</td>
<td>MAW</td>
<td>76.7</td>
<td>16.7</td>
<td>1.41</td>
</tr>
<tr>
<td></td>
<td>EWS</td>
<td>23.5</td>
<td>15.7</td>
<td>1.50</td>
</tr>
<tr>
<td></td>
<td>EWA</td>
<td>23.5</td>
<td>16.7</td>
<td>1.41</td>
</tr>
<tr>
<td></td>
<td>LMS</td>
<td>19.6</td>
<td>1.20</td>
<td>14.1</td>
</tr>
<tr>
<td>User Work-Station</td>
<td>MAW</td>
<td>445.9</td>
<td>52.7</td>
<td>5.22</td>
</tr>
<tr>
<td></td>
<td>EWS</td>
<td>275.2</td>
<td>59.5</td>
<td>4.63</td>
</tr>
<tr>
<td></td>
<td>EWA</td>
<td></td>
<td>52.1</td>
<td>5.28</td>
</tr>
<tr>
<td></td>
<td>LMS</td>
<td></td>
<td>53.0</td>
<td>5.19</td>
</tr>
</tbody>
</table>
Conclusions

- DVS is very effective for energy reduction
  - Upto 2 orders of magnitude savings possible
  - About 30% ‘instantaneous’ performance loss
- Averaged workloads are best
  - Makes system sluggish to workload changes
  - Unknown a priori
- Energy Performance Tradeoff
  - Faster updates lower visible performance loss
  - Faster updates also mean increased energy
- Workload prediction is crucial
- Adaptive LMS filtering is quite effective